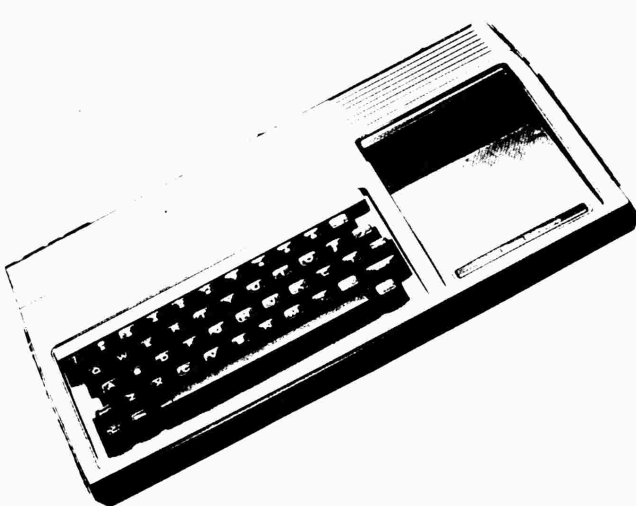


PARCO
Electrics

99/4A

99/4A



**INDEPENDENT MAGAZINE FOR THE
USERS OF THE T.I. 99/4A HOME
COMPUTER**

**VOLUME 2
ISSUE 4**

LETTERS LETTERS LETTERS LETTERS LETTERS

LETTERS

THE ONE THAT GOT AWAY.

S. SAWYER OF LINCOLN HAS NOW
CLAIMED A HIGH SCORE FOR
POPEYE OF 193,110. WE FOUND
YOUR LETTER AFTER WE HAD
PRINTED THE HIGH SCORES
MR SAWYER BUT FEAR NOT IT
WILL BE IN IT'S CORRECT
PLACE IN THE NEXT MAG

SORRY ED

SAM NASH WRITES FROM LEEDS.

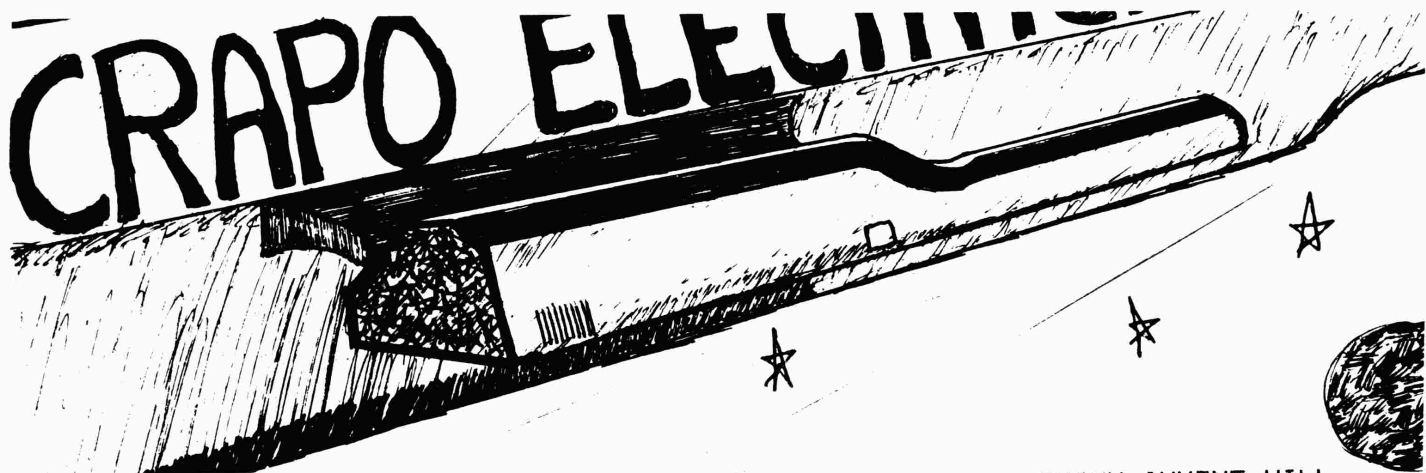
WHATEVER HAPPENED TO THE IDEA
OF PUTTING THE LISTED PROGRAMS
ON A CASSETTE? I'M ABSOLUTELY
SURE IT'S A GOOD IDEA

WE THINK SO TO SAM. SEE PAGE
9 ON HOW TO ORDER,
ed.

MANUEL MENDES FROM PORTUGAL WRITES
HERE IS THE PROGRAM THAT STEPHEN
MEADOWS ASKED IN Vol 2 Issue 3
FOR MOVING SPRITES AT ANY ANGLE

```
10 CALL CLEAR
20 CALL CHAR(33,"0008081C7F1C080
8")
30 INPUT "ANGLE "":ANG
40 INPUT "SPEED "":SP
50 CALL CLEAR
60 ANGLE=ANG*(PI/180)
70 CALL SPRITE(#1,33,5,96,124)
80 CALL MOTION(#1,-SIN(ANGLE)*SP
,COS(ANGLE)*SP)
90 GOTO 80
```

NICE ONE MANUEL
ed.



WHAT DO YOU MEAN 7 MONTHS LATE WITH THE MAGAZINE. DO YOU THINK ANYONE WILL
HAVE SPOTTED THE DELAY? SHOULD WE SHOOT THE EDITOR OR LEAVE HIM HIDING IN
THE DUSTBIN OUTSIDE THE OFFICES. PERHAPS WE COULD CONVINCE PEOPLE THAT IT
WAS THE DOUBLE WHARP DRIVE OR THE BACK PEDALLING MUD PUMPS THAT CAUSED ALL
THE TROUBLE. AT THAT MOMENT THE EDITOR WALKED BACK INTO THE ROOM. WE MUST
GET A MAGAZINE OUT NOW HE CRIED. I'D NEVER SEEN A GROWN MAN CRYING, IT WAS
HORRIBLE. QUICK THINK UP SOME STORY ABOUT THE DELAY FRANKLY ASKED. WE CAN
PIN IT ON THE POST OFFICE STRIKE LAST WEEK SAID RICK SHAW. WHOS RICK SHAW?
HE'S A CHINESE TAXI DRIVER. WHATS A CHINESE TAXI DRIVER DOING ON OUR SHIP?
..HES LOST. SO AM I THE STRIKE ONLY LASTED TWO DAYS.ANY OTHER IDEAS THEN?
.....OK WE SHOOT THE EDITOR.... BUT WAIT I HERE THE PRESSES RUNNING,,,,,
THEY HAVE HEARD OF THE TROUBLE TOO HAVE THEY?. DON'T PANIC I HAVE A GREAT
SOLUTION. BUT BEFORE THE SOLUTION WAS GIVEN THIS PAGE WAS SNATCHED, STUCK
ONTO THE BOTTOM OF PAGE TWO AND BEFORE YOU COULD SAY ** CRAPO ELECTRICA **
THE PRESSES WERE PRINTING VOLUME 2 NO 4. AND ABOUT TIME TOO I HERE YOU
SAY !

99/4A

**VOLUME 2
ISSUE 4**

CONTENTS:

PAGE

2	LETTERS
2	CRAPO ELECTRICA
4	COMPETITION
5	SOUND ADVICE
9	KEYBOARD KRUNCHERS
17	TI BITS
18	REVIEW - MUSIC MAKER
20	HELP - ON ADVENTURES
22	BRIGHT SPARKS
29	KIDS CORNER
32	HIGH SCORES & TOP FIVE

© 1987 PARCO ELECTRICS

99 4A MAGAZINE

IS PRODUCED BY

PARCO ELECTRICS

1. Manor Close, Weston, Honiton, Devon.

GREAT COMPETITION

BELOW ARE A FEW WORDS AVAILABLE IN THE SPEECH SYNTHESIZER VOCABULARY. DELETE ALL THE WORDS FROM THE GRID, THEN UNSCRAMBLE THE REMAINING LETTERS TO FIND TWO WORDS. THE FIRST THREE CORRECT ENTRIES WILL WIN 20 POUNDS WORTH OF SOFTWARE OF THEIR CHOICE.

A	C	O	N	N	E	C	T	E	D	N	E
Y	O	O	S	O	I	T	N	E	N	V	X
E	M	S	M	T	O	N	E	S	L	L	A
S	M	E	N	P	O	U	E	E	M	O	C
I	A	U	R	E	U	U	W	N	A	H	T
N	D	I	S	K	E	T	T	E	T	H	L
E	N	O	D	S	M	A	E	O	E	A	Y
T	O	O	H	E	A	X	B	R	R	V	O
V	W	C	I	R	T	V	E	O	E	E	U
P	A	R	T	N	E	R	E	T	U	D	Z
E	Y	E	R	E	H	T	E	G	O	T	T

ABOUT
ALL
ASSUME
BETWEEN
BOTH
COME
COMMA
COMPUTER
CONNECTED
DISKETTE

END
EXACTLY
EYE
HAVE
HIT
LET
MEAN
NINE
NOW
DONE

RED
SAID
SAVE
TEN
THAN
THERE
TOGETHER
PARTNER
TONE
TWELVE

YES
YOU
ZERO
TYPE
TOO
TO
EACH
PRINT
OUT

**** ODD ONE OUT **** RESULTS FROM VOL 2. No. 3 ****

THERE WERE TOO MANY ANSWERS BETTER THAN OURS, WHICH WE HAD FORGOTTEN ANYWAY. SO WE HAVE AWARDED PRIZES TO THE FIRST FIVE PLAUSIBLE ANSWERS RECEIVED...

TEN POUND VOUCHER WINNERS

D.ARMER./S.CROSS./P.HUTCHISON./T.WILDE./M MENDES

SOUND ADVICE

- Extended Basic

```

100 REM MUSIC PART 4
110 REM BY PHIL DONALD
120 REM FOR PARCO ELECTRICS
130 CALL MAGNIFY(3)
140 CALL CLEAR :: CALL SCREEN(16)
150 DISPLAY AT(4,13):"MUSIC" ::
160 DISPLAY AT(6,14):"BY" :: DISPLAY
170 AT(8,10):"PHIL DONALD" :: DISPL
180 AY AT(10,12):"PART 4" :: GOSUB
190 1210
150 CALL CLEAR :: DISPLAY AT(2,1
160 0):"MENU" :: DISPLAY AT(3,10):"-
170 ---" :: DISPLAY AT(12,1):"1) KEY
180 SIGNATURES" :: DISPLAY AT(14,1)
190 : "2) BAR LINES DEMONSTRATION"
160 DISPLAY AT(16,1):"3) TIME SI
170 GNATURES" :: DISPLAY AT(18,1):"4
180 ) END"
170 CALL KEY(0,K,S):: IF S=0 THE
180 N 170 :: K=K-48 :: IF K<1 OR K>4
190 THEN 170 :: ON K GOTO 190,700,8
200 10,180
180 CALL CLEAR :: FOR S=1 TO 8 :
210 : CALL SOUND(S,S*42*S*4,1):: NEX
220 T S :: STOP
190 CALL CLEAR :: DISPLAY AT(12,
230 1):"PRESS E FOR EXPLANATION OF
240 KEY SIGNATURES OR PRESS ANY OTHE
250 R KEY FOR LIST." :: FOR CH=96 TO
260 109 :: CALL CHAR(CH,""):: NEXT
270 CH
280 CALL CHAR(35,"0028287C287C28
290 28")
210 CALL KEY(0,K,S):: IF S=0 THE
220 N 210 :: IF K>69 THEN 470
230 CALL CLEAR :: DISPLAY AT(12,
240 1):"KEY SIGNATURES ENABLE YOU TO
250 SEE, AT A GLANCE, WHICH KEY YOU
260 ARE WORKING IN." :: GOSUB 1210 :
270 : GOSUB 1220
230 DISPLAY AT(12,1):"ONE SHARP
240 TELLS YOU THAT THE KEY IS G MAJOR
250 , TWO SHARPS D MAJOR, THREE SHAR
260 PS A MAJOR AND SO ON. BY COUNTIN
270 G FROM G TO D YOU WILL FIND THAT

```

240 DISPLAY AT(17,1):"THERE IS A
GAP OR INTERVAL OF FIVE OR A F
IFTH G,A,B,C,D AND BY COUNTING F
ROM D TO A YOU WILL SEE THAT THER
E IS AN INTERVAL OF A FIFTH HER
E"

250 DISPLAY AT(22,1):"ALSO." ::
GOSUB 1210 :: GOSUB 1220 :: DISP
LAY AT(12,1):"D,E,F,G,A. USING T
HIS METHOD YOU WILL BE ABLE TO WO
RK OUT HOW MANY SHARPS ARE IN THE
"

260 DISPLAY AT(15,1):"NEXT KEY A
ND ALSO THE NAME OF THE KEY. T
HIS METHOD IS CALLED THE CIRCLE
OF FIFTHS BUT WRITING THE KEY AN
D NAME OF THE SHARPS LEAVES YOU"

270 DISPLAY AT(20,1):"WITH THE R
IGHT HAND SIDE OF A PYRAMID." ::

GOSUB 1210 :: CALL CLEAR :: DIS
PLAY AT(1,4):"CIRCLE OF FIFTHS"

:: DISPLAY AT(3,1):"KEY SHARPS"

280 RESTORE 290 :: FOR R=1 TO 8
:: READ A,B,TEXT\$:: DISPLAY AT(
A,B):TEXT\$:: NEXT R :: CALL CHA
R(102,"4040405E62424C70")

290 DATA 6,1,C,8,1,G F#,10,1,D
F# C#,12,1,A F# C# G#,14

,1,E F# C# G# D#,16,1,B F
C# G# D# A#

300 DATA 18,1,F# F# C# G# D#
A# E#,20,1,C# F# C# G# D#

A# E# B#

310 GOSUB 1210 :: CALL CLEAR ::
DISPLAY AT(12,1):"ONE WAY OF REM

EMBERING THE SHARP NOTES IN ORD
ER IS TO MAKE UP A LITTLE SAYIN

G FATHER CHARLES GOES DOWN A
ND EATS BREAD"

320 GOSUB 1210 :: CALL CLEAR ::
DISPLAY AT(12,1):"REMEMBER, ALSO

, FROM THE MAJOR SCALES THAT
THE DISTANCE BETWEEN THE T

HIRD AND FOURTH NOTES IS A"

330 DISPLAY AT(16,1):"SEMITONE A
ND BETWEEN THE SEVENTH AND EI

GHTH NOTES IS ALSO A SEMITONE.
THE WHOLE THEORY SHOULD BECOME C

LEARER" :: GOSUB 1210 :: CALL CL
EAR

340 DISPLAY AT(12,1):"NOW YOU MA
Y BE WONDERING WHERE THE FLAT

NOTES COME FROM. WELL THE ME
THOD IS PRACTICALLY THE SAME E
XCEPT WE USE THE CIRCLE OF FOUR
THS"

```

350 GOSUB 1210 :: CALL CLEAR ::
DISPLAY AT(12,1):"C MAJOR HAS NO
FLATS OR SHARPS IN ITS KEY
SIGNATURE BUT WE CAN USE IT AS A
BASE TO FIND THE KEY A FOURTH
ABOVE IT"
360 DISPLAY AT(17,1):"SO BY COUN
TING UP WE WILL FIND C,D,E,F.
THE KEY WILL BE F MAJOR. THE F
OURTH NOTE OF THIS PARTICULAR KEY
HAS TO BE FLATTENED." :: GOSUB
1210
370 CALL CLEAR :: DISPLAY AT(12,
1):"TO FIND THE FOURTH OF THIS
KEY WE AGAIN COUNT UP THUS- F,G,
A,B. B IS THE NOTE TO BE FLATT
ENED. THE CIRCLE OFFOURTHS DIAG
RAM SHOULD"
380 DISPLAY AT(17,1):"HELP YOU.
ALL YOU NEED DO IS COUNT IN FO
URS....B,C,D,EAND SO ON." :: GOS
UB 1210 :: CALL CLEAR :: DISPLAY
AT(1,4):"CIRCLE OF FOURTHS"
390 DISPLAY AT(3,1):"KEY FLATS"
:: RESTORE 400 :: FOR R=1 TO 5 :
: READ A,B,TEXT$ :: DISPLAY AT(A
,B):TEXT$ :: NEXT R
400 DATA 6,1,C,8,1,F Bf,10,1,Bf
Bf Ef,12,1,Ef Bf Ef Af,14,1
Af Bf Ef Af Df
410 GOSUB 1210 :: CALL CLEAR ::
DISPLAY AT(12,1):"THERE ARE TWO
SCALES THAT WEHAVE NOT COVERED.
THESE ARE FLAT AND G FLAT MAJO
R. THE SCALES ARE MORE COMMON
LY"
420 DISPLAY AT(16,1):"KNOWN AS C
# MAJOR AND F# MAJOR RESPECTI
VELY." :: GOSUB 1210 :: CALL CLE
AR :: DISPLAY AT(12,1):"NOW THE
SHARPS AND FLATS AREKNOWN YOU WO
N'T BE SURPRISED"
430 DISPLAY AT(14,1):"TO LEARN T
HAT WHEN THESE AREPLACED IN THE
KEY SIGNATURE THEY ARE USED IN T
HE ORDER THAT THEY APPEAR IN ON
THE CIRCLE OF FIFTHS AND FOUR
THS"
440 DISPLAY AT(19,1):"CHARTS." :
: GOSUB 1210 :: CALL CLEAR :: DI
SPLAY AT(12,1):"NEVER MIX SHARPS
AND FLATS IN A KEY SIGNATURE,
THIS IS REGARDED AS BAD NOTATION
"
450 DISPLAY AT(15,1):"AND ABOVE
ALL ELSE IS RATHERCONFUSING. SH
ARPS AND FLATSCAN BE MIXED ON TH
E STAVE BUT WHEN THIS IS DONE
YOU WILL BE WORKING IN A MINOR
"

```

```

460 DISPLAY AT(20,1):"KEY NOT A
MAJOR." :: GOSUB 1210
470 CALL CLEAR :: DISPLAY AT(12,
1):"DEFINING CHARS NOW" :: GOSUB
1110
480 CALL DELSPRITE(ALL):: CALL C
LEAR :: PRINT "KEY SIG. PRESS
-----"
      G      G
      D      D
      A      A"
490 PRINT "      E      E
      B      B
      F SHARP FS
      C SHARP CS
      "
500 PRINT "-----"
      F      F
      B FLAT BF      E
      FLAT EF"
510 PRINT "      A FLAT AF
      "
      Q      QUIT": : :: DIS
PLAY AT(1,1):"CHOOSE KEY SIGNATU
RE"
520 ACCEPT AT(23,1)VALIDATE(VALP
HA)SIZE(2):K$ :: IF K$<"Q" THEN
530 ELSE CALL CHARSET :: GOTO 1
50
530 CALL CLEAR :: FOR R=2 TO 10
STEP 2 :: CALL HCHAR(R,6,33,26):
: NEXT R :: GOSUB 1170 :: IF K$<
>"G" THEN 550
540 CALL SPRITE(#1,120,5,5,50)::
DISPLAY AT(12,1):"KEY OF G MAJO
R" :: GOSUB 1210 :: GOTO 480
550 IF K$<"D" THEN 560 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,31
,70):: DISPLAY AT(12,1):"KEY OF
D MAJOR" :: GOSUB 1210 :: GOTO 4
80
560 IF K$<"A" THEN 570 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,31
,70,#3,120,5,1,90):: DISPLAY AT(
12,1):"KEY OF A MAJOR" :: GOSUB
1210 :: GOTO 480
570 IF K$<"E" THEN 590 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,31
,70,#3,120,5,1,90,#4,120,5,21,11
0):: DISPLAY AT(12,1):"KEY OF E
MAJOR" :: GOSUB 1210
580 GOTO 480
590 IF K$<"B" THEN 610 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,31
,70,#3,120,5,1,90,#4,120,5,21,11
0,#5,120,5,46,130)
600 DISPLAY AT(12,1):"KEY OF B M
AJOR" :: GOSUB 1210 :: GOTO 480
610 IF K$<"FS" THEN 630 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,3
1,70,#3,120,5,1,90,#4,120,5,21,1
10,#5,120,5,46,130)

```

```

620 CALL SPRITE(#6,120,5,12,150)
:: DISPLAY AT(12,1):"KEY OF F SH
ARP MAJOR" :: GOSUB 1210 :: GOTO
480
630 IF K$<>"CS" THEN 650 :: CALL
SPRITE(#1,120,5,5,50,#2,120,5,3
1,70,#3,120,5,1,90,#4,120,5,21,1
10,#5,120,5,46,130)
640 CALL SPRITE(#6,120,5,12,150,
#7,120,5,37,170):: DISPLAY AT(12
,1):"KEY OF C SHARP MAJOR" :: GO
SUB 1210 :: GOTO 480
650 IF K$<>"F" THEN 660 :: CALL
SPRITE(#1,124,5,37,50):: DISPLAY
AT(12,1):"KEY OF F MAJOR" :: GO
SUB 1210 :: GOTO 480
660 IF K$<>"BF" THEN 670 :: CALL
SPRITE(#1,124,5,37,50,#2,124,5,
12,70):: DISPLAY AT(12,1):"KEY O
F B FLAT MAJOR" :: GOSUB 1210 ::
GOTO 480
670 IF K$<>"EF" THEN 680 :: CALL
SPRITE(#1,124,5,37,50,#2,124,5,
12,70,#3,124,5,46,90):: DISPLAY
AT(12,1):"KEY OF E FLAT MAJOR" :
: GOSUB 1210 :: GOTO 480
680 IF K$<>"AF" THEN 480 :: CALL
SPRITE(#1,124,5,37,50,#2,124,5,
12,70,#3,124,5,46,90,#4,124,5,21
,110)
690 DISPLAY AT(12,1):"KEY OF A F
LAT MAJOR" :: GOSUB 1210 :: GOTO
480
700 CALL CLEAR
710 CALL CHAR(34,RPT$("10",8),35
,"02020202020E1E0C",36,"10101010
1070906",37,"1F1010101070F16",38
,"E020202020E0E0C",33,"000000000
OFF")
720 DISPLAY AT(12,1):"BAR LINES
ARE USED TO SEPERATE THE M
USIC INTO THAT OF THE TIME SIGNAT
URE." :: CALL HCHAR(5,1,33,32)::
CALL VCHAR(4,4,34,3)
730 FOR C=6 TO 12 STEP 2 :: CALL
HCHAR(5,C,35):: NEXT C :: CALL
VCHAR(4,14,34,3):: CALL HCHAR(5,
16,36):: CALL HCHAR(5,20,35):: C
ALL HCHAR(5,22,37)
740 CALL HCHAR(5,23,38):: CALL V
CHAR(4,25,34,3):: CALL HCHAR(5,2
7,36):: CALL HCHAR(5,29,36):: CA
LL VCHAR(4,31,34,3):: GOSUB 1210
:: DISPLAY AT(4,1)SIZE(1):"4"
750 DISPLAY AT(6,1)SIZE(1):"4" :
: DISPLAY AT(12,1):"IF YOU HAVE
A TIME OF 4/4, AS IN THE ABOVE
EXAMPLE, THEN AFTER THOSE FOUR
R BEATS YOU PLACE A BAR LINE."
760 DISPLAY AT(17,1):"IF YOU HAV
E A TIME OF 3/4 THEN THE BAR I
S PLACED AFTER THE THIRD BEAT AND
SO ON." :: GOSUB 1210 :: GOSUB
1220

```

```

770 DISPLAY AT(12,1):"THERE IS A
SHORT LIST OF COMMON SIMPLE
TIMES IN THIS PROGRAM." :: GOSUB
1210 :: GOSUB 1220 :: DISPLAY A
T(12,1):"LISTEN TO THE ABOVE EXA
MPLE"
780 DISPLAY AT(13,1):"AND TRY TO
FOLLOW THE NOTES" :: FOR R=1 TO
4 :: CALL SOUND(500,262,R):: NE
XT R :: CALL SOUND(1000,262,0)::
CALL SOUND(500,262,1)
790 FOR R=1 TO 2 :: CALL SOUND(2
50,262,R+1):: NEXT R :: FOR R=0
TO 1 :: CALL SOUND(1000,262,R)::
NEXT R :: DISPLAY AT(20,1):"PLA
Y TUNE AGAIN Y/N"
800 CALL KEY(0,K,S):: IF S=0 THE
N 800 :: IF K=89 THEN 780 ELSE 1
50
810 CALL CLEAR :: DISPLAY AT(12,
1):"THE FOLLOWING LIST OF SIMPLE
TIME SIGNATURES WILL GIVE YOU
AN UNDERSTANDING OF HOW NOTES SH
OULD BE GROUPED AND"
820 DISPLAY AT(16,1):"WHERE THE
BEATS, OR PULSES, ARE. EACH TIM
E SIGNATURE CHOSEN WILL BE ACC
OMPANIED BY A SHORT TUNE." :: G
OSUB 1210 :: GOSUB 1220
830 DISPLAY AT(12,1):"IDEALLY, Y
OU SHOULD BE ABLE TO LOOK AT A P
IECE OF MUSIC AND KNOW WHAT THE
TIME SIGNATURE IS JUST BY L
OOKING AT HOW THE NOTES ARE GROUP
ED."
840 DISPLAY AT(17,1):"THIS OF CO
URSE TAKES A BIT OF PRACTICE SO
WE WILL LOOK AT WHAT THE NUMBER
S MEAN FIRST." :: GOSUB 1210
:: GOSUB 1220
850 DISPLAY AT(12,1):"IN SIMPLE
TIME THE FIRST NUMBER IS THE
NUMBER OF BEATS PER BAR AND
THE SECOND NUMBER IS THE TYPE OF
BEAT."
860 DISPLAY AT(17,1):"4/4 FOUR
CROTCHET BEATS PER BAR
2/2 TWO MINIM BEA
TS PER BAR
3/8 THREE QUAVER BEATS PE
R BAR"
870 GOSUB 1210 :: GOSUB 1220 ::
DISPLAY AT(12,1):"OBVIOUSLY YOU
DON'T HAVE TO FILL A BAR IN 4/4
TIME WITH FOUR CROTCHETS. YOU C
AN USE SIXTEEN SEMIQUAVERS IF YO
U"
880 DISPLAY AT(16,1):"LIKE BUT T
HERE MUST BE FOUR PULSES OTHERWI
SE YOU WON'T BE COMPOSING IN 4/
4 SIMPLE TIME." :: GOSUB 1210 :
: GOSUB 1220

```

```

890 DISPLAY AT(12,1):"DEFINING C
HARS NOW" :: RESTORE 900 :: FOR
R=1 TO 16 :: READ CH,CH$ :: CALL
CHAR(CH,CH$):: NEXT R
900 DATA 96,02020202020E1E0C,97,
0C0A0A0A0838783,98,1010101010709
06,99,1F1010101070F16,100,E02020
2020E0E0C,101,FF20202020E1E3C1
910 DATA 102,C040404040C0C08,103
,FF40404040C3C783,104,8080808080
808,105,1F101F101070F16,106,FF20
FF2020E1E3C1,107,FF40FF4040C3C78
3
920 DATA 108,00122E04040808,109,
080808080838793,33,0000000000FF,
34,808080808080808
930 CALL CLEAR
940 PRINT : : : "TIME SIG.  PR
ESS

```

4/4 44

```

3/4 34"
950 PRINT : : " 999 QUIT
": : : : : ACCEPT AT(22,1)VALID
ATE(DIGIT):KS :: CALL CLEAR :: I
F KS=999 THEN 150
960 IF KS<>44 THEN 1030
970 RESTORE 980 :: FOR R=1 TO 16
:: READ A,B,C,D :: CALL HCHAR(A
,B,C,D):: NEXT R
980 DATA 5,1,33,32,5,5,96,1,5,8,
96,1,5,12,96,1,5,16,96,1,5,20,96
,1,5,24,96,1,5,28,96,1
990 DATA 8,1,33,32,8,4,96,1,8,8,
108,1,8,10,97,1,8,12,96,1,8,16,1
08,1,8,18,97,1,8,20,96,1
1000 DISPLAY AT(4,1)SIZE(1):"4"
:: DISPLAY AT(6,1)SIZE(1):"4" ::
CALL VCHAR(4,4,34,3):: CALL VCH
AR(4,18,34,3):: CALL VCHAR(7,6,3
4,3):: CALL VCHAR(7,24,34,3)
1010 FOR R=1 TO 6 :: CALL SOUND(
500,494,1):: NEXT R :: CALL SOUN
D(500,440,1):: CALL SOUND(500,41
5,1):: CALL SOUND(250,44000,30):
: CALL SOUND(250,370,1)
1020 CALL SOUND(500,440,1):: CAL
L SOUND(250,44000,30):: CALL SOUN
D(250,330,1):: CALL SOUND(500,4
15,1):: GOSUB 1230 :: GOTO 1010
1030 IF KS<>34 THEN 930
1040 RESTORE 1050 :: FOR R=1 TO
15 :: READ A,B,C,D :: CALL HCHAR
(A,B,C,D):: NEXT R
1050 DATA 5,1,33,32,5,5,96,1,5,8
,96,1,5,12,96,1,5,18,96,1,5,22,9
6,1,5,26,96,1,8,1,33,32
1060 DATA 8,4,96,1,8,8,96,1,8,12
,96,1,8,18,96,1,8,22,99,1,8,23,1
00,1,8,25,96,1
1070 CALL VCHAR(4,4,34,2):: CALL
VCHAR(4,16,34,2):: CALL VCHAR(4
,30,34,2):: CALL VCHAR(7,16,34,3
):: CALL VCHAR(7,29,34,3)

```

```

1080 DISPLAY AT(4,1)SIZE(1):"3"
:: DISPLAY AT(6,1)SIZE(1):"4" ::
CALL SOUND(500,330,1):: CALL SOUN
D(500,440,1):: CALL SOUND(500,
659,1):: CALL SOUND(500,392,1)
1090 CALL SOUND(500,659,1):: CAL
L SOUND(500,440,1):: CALL SOUND(
500,370,1):: CALL SOUND(500,440,
1):: CALL SOUND(500,659,1):: CAL
L SOUND(500,349,1)
1100 CALL SOUND(250,587,1):: CAL
L SOUND(250,554,1):: CALL SOUND(
500,440,1):: GOSUB 1230 :: GOTO
1080
1110 RESTORE 1120 :: FOR R=1 TO
28 :: READ CH,CH$ :: CALL CHAR(C
H,CH$):: NEXT R :: CALL CHAR(120
,"2424FF242424FF2424"&RPT$("0",5
0),124,"4040405E62424C70"&RPT$("
0",52)):: RETURN
1120 DATA 33,FF,34,0000010102020
404,35,080808101010102,36,202020
404040404,37,404040202010100C,38
,0201,39,01060404040403
1130 DATA 42,00000001010101,43
,02020202020202,135,030306060A
122242,58,42820202020202,59,02
020202031E2222,60,42424242424242
62
1140 DATA 61,1E020202020202,62
,02FF010101010101,63,01010101010
101FE,64,000000001E214040,91,808
08,92,00010106081060C,93,8
1150 DATA 94,00000000FC03,95,000
0000000010204,96,18E,97,00000000
0000804,98,404040404040808,99,00
0000000000C02,100,10101010101010
2
1160 DATA 101,202040408
1170 RESTORE 1180 :: FOR R=1 TO
28 :: READ RX,CX,DX :: CALL HCHA
R(RX,CX,DX):: NEXT R :: RETURN
1180 DATA 5,2,34,6,2,35,7,2,36,8
,2,37,9,2,38,10,2,39,2,3,42,3,3,
43,4,3,135,5,3,58,6,3,59,7,3,60
1190 DATA 8,3,61,9,3,62,10,3,63,
1,4,64,2,4,91,3,4,92,4,4,93,6,4,
94,8,4,95,9,4,96,1,5,97,2,5,98
1200 DATA 3,5,93,6,5,99,7,5,100,
8,5,101
1210 DISPLAY AT(24,9):"PRESS ANY
KEY" :: CALL KEY(0,K,S):: IF S=
0 THEN 1210 :: RETURN
1220 FOR R=12 TO 24 :: DISPLAY A
T(R,1):" " :: NEXT R :: RETURN
1230 DISPLAY AT(24,1):"PLAY TUNE
AGAIN Y/N" :: CALL KEY(0,K,S)::
IF S=0 THEN 1230 :: IF K=89 THE
N RETURN ELSE 930

```

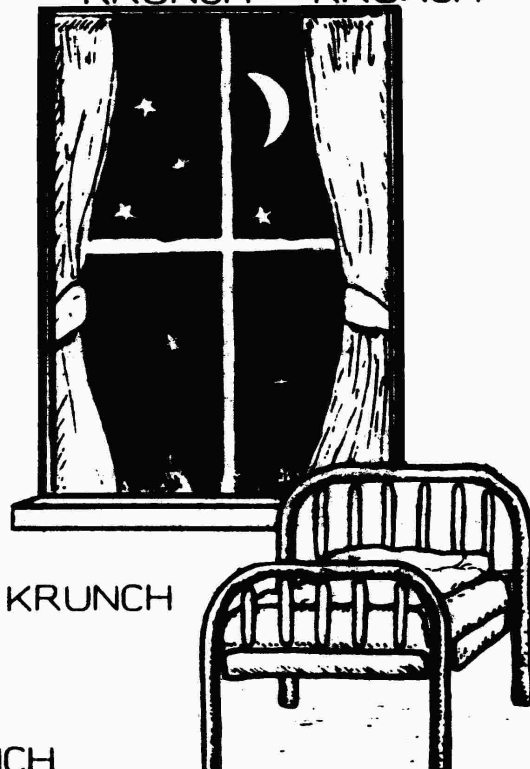
KEYBOARD KRUNCHERS

THE LIST *****

PAGE ****	TITLE *****
9	ALIEN BY- M.MENDES
14	TRENCH ATTACK BY- P.HUTCHISON

BOTH PROGRAMS IN EXT. BASIC

KRUNCH * KRUNCH



KRUNCH

★ ALIEN ★

YOU MUST PASS 3 SCREENS
TO PICK 1 PERON ROCK
WHICH IS ESSENTIAL FOR
YOUR PLANETS SURVIVAL.
THE MISSION WILL BE
COMPLETED AFTER YOU HAVE
COLLECTED 5 PERON ROCKS.

--FIRST SCREEN--

★*ALIEN ATTACK*★

KEYS S = LEFT
D = RIGHT
SPACE BAR = TO JUMP
Q = TO FIRE

YOU MUST HIT A NUMBER OF
FLYING SAUCERS

--SECOND SCREEN--

▶* IN SPACE *◀

KEYS E = UP
X = DOWN
Q = FIRE

--THIRD SCREEN--

★*ALIEN HOMEBASE*★

KEYS S = LEFT
D = RIGHT
Q = FIRE

KRUNCH
KRUNCH * KRUNCH * KRUNCH

```

100 RANDOMIZE
110 ! INITIAL COLORS
111 FOR E=1 TO 4 :: CALL COLOR(E
,11,1):: NEXT E :: FOR T=5 TO 8
:: CALL COLOR(T,7,1):: NEXT T
120 CALL CLEAR :: CALL SCREEN(2)
:: CALL MAGNIFY(3)
128 REM ** INITIAL CHARCTERS **
129 CALL CHAR(45,"0000003C3C0000
00")
131 CALL CHAR(64,"7E819DA0A09D81
7E")
140 FOR T=65 TO 90
141 CALL POSITION(1,X,Y):: IF Y
>=240 THEN 1390
150 READ A$
160 CALL CHAR(T,"00"&A$)
170 NEXT T
180 DATA FEC6C6FEC6C6C6,FEC6C6FC
C6C6FE,FEC2C0C0C0C0C2FE,FC6666E666
66FC,FEC0C0FCC0C0FE,FEC0C0FEC0C0
C0,FCC6C0CEC6C6FC,C6C6C6FEC6C6C6
190 DATA 3C18181818183C,1E0C0C0C
CCCC78,C6DCF0E0F0DC0C,C0C0C0C0C0
C0FE,C6EED6D6C6C6C6,C6E6F6DECEC6
C6,FEC6C6C6C6C6FE,FEC6C6FEC0C0C0
200 DATA FEC6C6C6D6CEFE,FCC6C6FC
C6C6C6,FEC0C0FE0606FE,FC30303030
3030,C6C6C6C6C6C6FE,82C6C666C6C3
38,C6C6C6D6D6EEC6,C66C3838386C06
,82C66C38383838
    
```

```

210 DATA FE060C183060FE
220 CALL CHAR(128,"010101030F3F7
7F7F7F3E1C0000000000808080C0F0FCF
EFEFE7C3800000000000")
320 REM ** EXPLOSION1 **
330 CALL CHAR(136,"081C1F6F3E7D1
AFD7A3D7A0D3F0B13014062ECFDF5EB
F5EBCBCACFEBDA88400")
340 REM ** EXPLOSION 2 **
350 CALL CHAR(140,"000000100A0F3
D1A751A3D7F0F020000000000090E0C07
8B040B040C020100000")
351 REM ** INICIALIZAR **
352 MI=0 :: LA=100 :: PER=0 :: L
IVES=5
360 CALL CHAR(46,"00000000080000
0000")
370 CALL CHAR(91,"00000010381000
00")
380 CALL CHAR(96,"010107093F7FFF
FF")
390 CALL CHAR(97,"00C0209088E4F2
FF")
400 CALL CHAR(98,"00000103075FFF
FF")
410 CALL CHAR(99,"80804040A8A8BD
FF")
420 CALL CHAR(100,"101010387474F
6FF",104,"0008142A00000000",105,
"00000018247E5A00",106,"003C42BD
A5000000",107,"0000000000000000"
)
421 CALL CHAR(108,"FF0000FF0000F
FFF",109,"007E7E7E7E7E00FF",110,
RPT$("93",16),111,RPT$("FC",16))
422 CALL CHAR(132,"01030301"&RPT
$("0",24)&"80C0C080"&RPT$("0",24
))
427 GOSUB 710
428 GOSUB 450
432 GOTO 1000
450 REM ** BACKGROUND SCREEN **
455 CALL CLEAR
460 DISPLAY AT(12,5):"WAIT A MOM
ENT PLEASE" :: FOR T=1 TO 500 ::
NEXT T :: CALL CLEAR
461 CALL COLOR(2,1,1,8,1,1)
470 FOR T=1 TO 100 :: CALL HCHAR
(INT(RND*23)+1,INT(RND*31)+1,46)
:: NEXT T
480 FOR T=1 TO 6 :: CALL HCHAR(I
NT(RND*24)+1,INT(RND*31)+1,91)::
NEXT T
490 A$=""`abcdbc`addbdcdb`adbc`b
d`acdbcd"
500 CO=22 :: GOSUB 550
510 A$="hkkikkkkhkkikkkikkkikkk
hkkikkkhi" :: CO=23 :: GOSUB 550
520 A$="kkjkkkjkkkjkkkjkkkjkkjkk
kkkkkjkk" :: CO=24 :: GOSUB 550
540 GOTO 566
550 FOR T=1 TO LEN(A$):: CALL HC
HAR(CO,0+T,ASC(SEG$(A$,T,1)))::
NEXT T
560 RETURN
566 RETURN

```

```

700 REM ** MAIN SCREEN **
710 CALL CLEAR
730 A$="ALIEN" :: B=1 :: K=10 ::
LIN=80
731 CALL SPRITE(#1,128,4,169,16,
0,40)
740 CALL POSITION(#1,CO,LI)
750 IF LI>LIN THEN 760 ELSE 740
760 CALL MOTION(#1,0,0):: CALL S
PRITE(#2,132,11,169,LI+1,-100,0)
770 CALL POSITION(#2,C,D):: IF C
>32 THEN 770
780 CALL MOTION(#2,0,0):: CALL S
OUND(220,-7,0):: CALL PATTERN(#2
,136):: CALL SOUND(220,-6,5):: C
ALL PATTERN(#2,140):: CALL SOUND
(400,-5,5)
790 CALL DELSPRITE(#2):: DISPLAY
AT(4,K):SEG$(A$,B,1):: K=K+2 ::
IF B=LEN(A$) THEN 800 ELSE B=B+1
:: LIN=LIN+16 :: CALL MOTION(#1
,0,10)
796 GOTO 740
800 CALL DELSPRITE(#1):: DISPLAY
AT(8,3):"©1985 BY MANUEL MIGUEL
"
810 DISPLAY AT(11,10):"EXT BASI
C"
820 FOR T=1 TO 10 :: CALL SOUND(
-600,INT(RND*2000)+500,2,INT(RND
*2000)+500,2,INT(RND*2000)+500,2
):: NEXT T
860 RETURN
999 GOSUB 10000
1000 REM ** FIRST SCREEN **
1002 CALL CHAR(128,"010101030F3F
7F7F7F3E1C0000000000808080C0F0FC
FEFEFE7C380000000000")
1020 CALL CHAR(124,"2070F8705A55
555A555A5A5570F87020040E1F0EA5A
5AA5A5A5A5A5A0E1F0E04")
1040 CALL CHAR(120,"00000001071F
3FEAE83F1E070100000000000080E0F8
FCFAFFFC78E080000000")
1050 CALL CHAR(116,"00030000192C
6DDFFFD87F2A5D6A3F1B38FC7C18BCBA
BAFDFF6DFEABDDABFE6C")
1051 PA=INT(3+PER*2.2/1.2):: SPD
1=-5-PER :: LA=140 :: DVN=4+PER*
2
1059 B$="UFO'S" :: B=DVN :: GOSU
B 9000
1060 CO=65
1070 DC=6 :: LI=88
1080 CONT=5
1090 FOR T=CONT TO CONT+2 :: CAL
L SPRITE(#T,124,INT(RND*13)+3,CO
,LI):: LI=LI+32 :: NEXT T
1091 IF CONT=11 THEN 1115
1100 DC=DC+14 :: CONT=CONT+3 ::
CO=CO+22 :: LI=88+DC :: GOTO 109
0
1115 A=INT(RND*8)+4 :: FOR T=5 T
O 7 :: CALL MOTION(#T,0,A):: NEX
T T

```

```

1116 A=INT(RND*8)+-4 :: FOR T=8
TO 10 :: CALL MOTION(#T,0,A)::
NEXT T
1117 A=INT(RND*8)+4 :: FOR T=11
TO 13 :: CALL MOTION(#T,0,A):: N
EXT T
1118 MD=0
1119 CALL SPRITE(#3,120,15,44,24
2,0,SPD1)
1120 CALL SPRITE(#1,128,16,180,I
NT(RND*150)+20)
1121 CALL SPRITE(#4,116,15,178,2
42,0,-7)
1390 CALL COINC(ALL,D):: IF D TH
EN 3000
1391 DVNI=INT(RND*20)+1 :: IF DV
NI<>5 THEN 1396 :: CALL MOTION(#
3,0,0):: CALL POSITION(#3,A,B)::
B1=INT((B/8)+.9):: CALL VCHAR(7
,B1,111,15)
1392 CALL COINC(#1,180,B1*8-7,8,
E):: CALL VCHAR(7,B1,32,15):: CA
LL MOTION(#3,0,SPD1):: IF E THEN
GOSUB 6000
1396 CALL KEY(0,K,S):: CALL COIN
C(ALL,D):: IF D THEN 3000
1400 IF MD=0 THEN 1406 ELSE CALL
POSITION(#2,R,E):: IF R<41 THEN
CALL DELSPRITE(#2):: MD=0
1406 IF S=0 THEN CALL MOTION(#1,
0,0):: GOTO 1390
1407 CALL COINC(ALL,D):: IF D TH
EN 3000
1408 IF K=83 THEN CALL POSITION(
#1,X,Y):: IF Y<14 THEN CALL MOTI
ON(#1,0,0):: GOTO 1390 :: ELSE C
ALL MOTION(#1,0,-9):: GOTO 1390
1420 IF K=68 THEN CALL POSITION(
#1,X,Y):: IF Y>242 THEN CALL MOT
ION(#1,0,0):: GOTO 1390 :: ELSE
CALL MOTION(#1,0,9):: GOTO 1390
1430 IF K=81 THEN GOTO 1500
1440 IF K=32 THEN CALL MOTION(#1
,0,0):: GOTO 1600
1450 CALL MOTION(#1,0,0):: GOTO
1390
1500 IF MD=1 THEN 1390
1501 LA=LA-1 :: DISPLAY AT(4,9):
"-&STR$(LA)&" :: IF LA<0 THEN
AS="YOU RUN OUT OF LAZER" :: GO
TO 8000
1504 MD=1 :: CALL MOTION(#1,0,0)
:: CALL POSITION(#1,R,E):: CALL
SOUND(-110,300,2,400,4,500,6)::
CALL SPRITE(#2,132,14,R-5,E,-18,
0):: GOTO 1390
1599 REM ** JUMP **
1600 CALL DELSPRITE(#2):: CALL P
OSITION(#1,X,Y):: E=18 :: FOR T=
179 TO 163 STEP -2 :: E=E-2 :: C
ALL SOUND(-100,-7,E)
1605 CALL COINC(ALL,D):: IF D TH
EN GOSUB 6000 :: CALL LOCATE(#1,
180,Y):: GOTO 1120

```

```

1610 CALL LOCATE(#1,T,Y):: NEXT
T
1620 FOR T=1 TO 70 :: NEXT T
1630 FOR T=163 TO 180 STEP 2 ::
E=E+2 :: CALL SOUND(-100,-7,E)::
CALL COINC(ALL,D):: IF D THEN G
OSUB 6000 :: CALL LOCATE(#1,180,
Y):: GOTO 1120
1640 CALL LOCATE(#1,T,Y):: NEXT
T :: GOTO 1390
2999 REM ** CALL COINC **
3000 CALL MOTION(#2,0,0):: CALL
COINC(#3,#2,16,CDC):: IF CDC THE
N 3001 :: CALL COINC(#1,#4,16,CD
C):: IF CDC=0 THEN 3010 ELSE GOS
UB 6000 :: GOTO 1120
3001 CALL MOTION(#1,0,0):: CALL
DELSPRITE(#2):: CALL COLOR(#3,7)
:: E=800 :: CALL MOTION(#3,20,0)
3002 CALL SOUND(-100,E,0):: E=E-
16 :: CALL POSITION(#3,B,R):: IF
B>170 THEN CALL MOTION(#3,0,0):
: GOTO 3003 :: ELSE GOTO 3002
3003 CALL SOUND(-220,-7,0):: CAL
L PATTERN(#3,136):: CALL SOUND(-
220,-6,2):: CALL PATTERN(#3,140)
:: CALL SOUND(-500,-5,6)
3005 CALL DELSPRITE(#3):: DVN=DV
N-1 :: DISPLAY AT(2,15):STR$(DVN
)&" ";
3006 IF DVN=0 THEN 4000 ELSE GOT
O 1119
3009 GOTO 1119
3010 CALL MOTION(#2,100,0)
3013 CALL SOUND(-40,110,1,200,1,
-6,8):: CALL SOUND(-40,220,1,415
,1,-5,0)
3014 CALL POSITION(#2,E,R):: CAL
L COINC(#1,180,R,16,CDC):: IF CD
C=0 THEN CALL DELSPRITE(#2):: GO
TO 1390 :: ELSE CALL DELSPRITE(
#2):: GOSUB 6000
3015 GOTO 1120
4000 REM ** SCREEN 2 **
4010 GOSUB 10000
4020 LIVES=LIVES+1 :: MI=MI+5 ::
B$=" ASTE" :: AST=6 :: B=AST ::
DIM AS(9):: FOR T=3 TO 8 :: AS(
T)=124 :: NEXT T :: GOSUB 9000
4030 CALL CHAR(116,"000000000001
02050301"&RPT$("00",11)&"80C0E0C
0800000000000000")
4040 CALL CHAR(120,"0000000000103
0E151A0D0703000000000000000080C0
E070F0E0C0000000000000")
4050 CALL CHAR(124,"0001061B356A
D5AD56A151F0F07030000C0E0F8FCFC
7EBF7FFFFEFCFC0C08000")
4060 CALL CHAR(128,"4F50F6FFF0F7
77703F000000000000000008080FC3ECF
CF3EF80000000000000000")
4070 CALL CHAR(132,RPT$("0",32)&
"00000000301C30"&RPT$("0",16))

```

```

4080 FOR T=1 TO 4 :: DISPLAY AT(
11,3):"ASTEROID BELT COMING !!"
:: FOR E=1 TO 5 :: CALL SOUND(1,
880,0,890,0):: NEXT E :: DISPLAY
AT(11,3):RPT$(" ",23):: FOR R=
1 TO 100 :: NEXT R
4090 NEXT T :: SP=SPD1
4091 CALL SPRITE(#1,128,15,100,1
4)
4100 T=50 :: FOR E=3 TO 8 :: CAL
L SPRITE(#E,124,INT(RND*13)+3,T,
242,0,INT(RND*SPD1)-10):: T=T+17
:: NEXT E
4110 CALL KEY(0,K,S):: CALL COIN
C(ALL,Q):: IF Q THEN 4800
4111 IF MP<>0 THEN CALL POSITION
(#2,E,R):: IF R>100 THEN CALL DE
LSPRITE(#2):: MP=0
4115 CALL COINC(ALL,Q):: IF Q TH
EN 4800
4121 IF S=0 THEN CALL MOTION(#1,
0,0):: GOTO 4110
4130 IF K=69 THEN CALL POSITION(
#1,X,Y):: IF X<59 THEN CALL MOTI
ON(#1,0,0):: GOTO 4110 :: ELSE C
ALL MOTION(#1,-8,0):: GOTO 4110
4140 IF K=88 THEN CALL POSITION(
#1,X,Y):: IF X>145 THEN CALL MOT
ION(#1,0,0):: GOTO 4110 :: ELSE
CALL MOTION(#1,8,0):: GOTO 4110
4150 IF K=81 THEN IF MP=1 THEN 4
110 ELSE CALL MOTION(#1,0,0):: G
OTO 4170
4160 CALL MOTION(#1,0,0):: GOTO
4110
4170 LA=LA-1 :: DISPLAY AT(4,9):
"- "&STR$(LA)&" " :: IF LA<0 THEN
A$="YOU RUN OUT OF LAZER" :: GO
TO 8000
4171 MP=1 :: CALL SOUND(-200,-5,
0):: CALL POSITION(#1,X,Y):: CAL
L SPRITE(#2,132,14,X,28,0,9):: G
OTO 4110
4180 GOTO 4110
4800 MP=0 :: CALL COLOR(#2,2)::
CALL MOTION(#1,0,0,#2,0,SPD1-2)
4801 FOR T=3 TO 8 :: IF AS(T)=11
2 THEN 4802 ELSE CALL COINC(#2,#
T,14,D):: IF D THEN 4810
4802 NEXT T
4803 GOSUB 6000 :: GOTO 4110
4807 CALL DELSPRITE(#2):: GOTO 4
110
4810 CALL PATTERN(#T,136):: CALL
SOUND(-200,-6,0):: AS(T)=AS(T)-
4 :: IF AS(T)<>112 THEN 4820 EL
S E CALL DELSPRITE(#2):: AST=AST-1
:: DISPLAY AT(2,14):"- "&STR$(AS
T);
4811 CALL DELSPRITE(#T):: IF AST
=0 THEN 5000
4812 GOTO 4110

```

```

4820 CALL DELSPRITE(#2):: CALL P
ATTERN(#T,AS(T))
4821 GOTO 4110
5000 REM ** SCREEN 3 **
5001 GOSUB 10000
5015 LIVES=LIVES+1 :: ALI=PER+2
:: M=0 :: TIME=150 :: MI=MI+5
5020 B$="ALIEN" :: B=ALI
5021 GOSUB 9000
5025 CALL CHAR(116,"00050A172F5F
997DBF7F985C2F170A050040A0D0E8F4
92BCFAFC3274E8D0A040")
5030 CALL CHAR(120,RPT$("01",16)
&RPT$("80",16))
5035 CALL CHAR(132,"01030301"&RP
T$("0",24)&"80C0C080"&RPT$("0",2
4))
5040 CALL CHAR(128,"000000000101
01030F3F7F7F7F3E1C00000000008080
80C0F0FCFEFEFE7C3800")
5041 CALL SPRITE(#1,128,16,178,I
NT(RND*241)+1)
5045 CALL SPRITE(#3,116,INT(RND*
13)+3,60,INT(RND*240)+1)
5060 CALL COINC(ALL,D):: IF D TH
EN 5100 ELSE W=INT(RND*3+1):: ON
W GOTO 5065,5067,5071
5065 CALL MOTION(#3,0,90):: CALL
MOTION(#3,0,0):: CALL SOUND(-1,
-2,0):: GOTO 5071
5067 CALL MOTION(#3,0,-90):: CAL
L SOUND(-1,-2,0):: CALL MOTION(#
3,0,0)
5071 E=INT(RND*10)+1 :: IF E<>7
THEN 5075 ELSE CALL POSITION(#3,
E,R):: CALL SPRITE(#4,120,13,80,
R):: CALL SOUND(-220,-6,2):: CAL
L DELSPRITE(#4)
5072 CALL COINC(#1,178,R,10,D)::
IF D THEN GOSUB 6000 :: GOTO 50
75
5075 CALL COINC(ALL,D):: IF D TH
EN 5100 :: IF M=0 THEN 5077 ELSE
CALL POSITION(#2,X,Y):: IF X<56
THEN CALL DELSPRITE(#2):: M=0
5077 CALL KEY(0,K,S):: IF S=0 TH
EN CALL MOTION(#1,0,0):: GOTO 50
60
5090 IF K=83 THEN CALL MOTION(#1
,0,-90):: CALL MOTION(#1,0,0)::
GOTO 5060
5095 IF K=68 THEN CALL MOTION(#1
,0,90):: CALL MOTION(#1,0,0):: G
OTO 5060
5096 IF K=81 THEN IF M=1 THEN 50
60 ELSE 5150
5098 CALL MOTION(#1,0,0):: GOTO
5060
5100 CALL COINC(#2,#3,16,D):: IF
D THEN CALL SOUND(-220,-7,0)::
CALL DELSPRITE(#2):: CALL DELSPR
ITE(#3):: M=0 :: ALI=ALI-1 :: DI
SPLAY AT(2,14):"- "&STR$(ALI)&" "
:: IF ALI=0 THEN 5500 ELSE 5045

```

```

5150 M=1 :: LA=LA-1 :: DISPLAY A
T(4,9): "-"&STR$(LA)&" " :: IF LA
<0 THEN A$="YOU RUN OUT OF LAZER
" :: GOTO 8000
5152 CALL MOTION(#1,0,0):: CALL
POSITION(#1,X,Y):: CALL SPRITE(#
2,132,14,180,Y,-25,0):: CALL SOU
ND(-220,-5,2):: GOTO 5060
5500 FOR S=110 TO 510 STEP 20
5510 CALL SOUND(-400,S,0,620-S,0
):: NEXT S :: DISPLAY AT(12,1):"
GOOD! YOU PICKED UP 1 ROCK"
5600 PER=PER+1 :: MI=MI+10 :: LI
VES=LIVES+1 :: FOR T=1 TO 500 ::
NEXT T :: CALL HCHAR(12,1,32,29
):: IF MI=100 THEN 15000 ELSE GO
TO 999
5999 REM ** YOU LOSE A BASE**
6000 CALL MOTION(#1,0,0):: CALL
DELSPRITE(#2):: CALL SOUND(200,-
5,0):: CALL PATTERN(#1,136):: CA
LL SOUND(200,-6,2):: CALL COLOR(
#1,7):: CALL PATTERN(#1,140)
6002 CALL SOUND(600,-7,6):: CALL
COLOR(#1,16):: CALL PATTERN(#1,
136)
6010 MD=0 :: LIVES=LIVES-1 :: DI
SPLAY AT(2,23): "-"&STR$(LIVES)&"
" :: IF LIVES=0 THEN A$="YOU RU
N OUT OF LIVES" :: GOTO 8000
6012 FOR T=1 TO 100 :: NEXT T ::
CALL PATTERN(#1,128):: RETURN
7999 REM ** LOSE GAME ROUTINE **
8000 CALL DELSPRITE(ALL):: CALL
CLEAR :: DISPLAY AT(1,4):A$;
8001 CALL SOUND(220,110,0):: FOR
T=1 TO 200 :: NEXT T
8010 DISPLAY AT(4,4):"*** MISSION
FAILED **"
8020 CALL SOUND(220,123,0):: FOR
T=1 TO 200 :: NEXT T
8030 DISPLAY AT(8,12):"SORRY";
8040 CALL SOUND(220,147,0):: FOR
T=1 TO 200 :: NEXT T
8050 DISPLAY AT(11,4):"BETTER LU
CK NEXT TIME"
8060 CALL SOUND(230,165,0):: FOR
T=1 TO 200 :: NEXT T
8070 DISPLAY AT(14,7):"YOU COMPL
ETED "&STR$(MI)&"%";
8071 DISPLAY AT(16,7):" OF THE M
ISSION"
8080 CALL SOUND(-220,175,0)
8090 DISPLAY AT(18,5):"AND YOU P
ICKED "&STR$(PER)&" ROCKS OF
PERONYO"
8100 CALL SOUND(-220,196,0)
8111 DISPLAY AT(22,4):" WELL DO
NE ANYWAY";
8112 CALL SOUND(-220,220,0)
8113 GOTO 15030

```

```

9000 REM ** BOARD **
9005 DISPLAY AT(2,0):"PERON-"&ST
R$(PER)&" "&B$&"-"&STR$(B)&" LI
VES-"&STR$(LIVES);:: DISPLAY AT(
4,0):"LAZ LEFT-"&STR$(LA)&" MISS
ION-"&STR$(MI)&"% ";
9006 CALL HCHAR(1,2,108,27):: CA
LL HCHAR(3,2,108,27):: CALL HCHA
R(5,2,108,27):: CALL VCHAR(1,1,1
10,5):: CALL VCHAR(1,29,110,5)
9010 CALL HCHAR(1,1,109):: CALL
HCHAR(3,1,109):: CALL HCHAR(5,1,
109):: CALL HCHAR(1,29,109):: CA
LL HCHAR(3,29,109):: CALL HCHAR(
5,29,109)
9030 CALL COLOR(0,8,8):: FOR E=1
TO 4 :: CALL COLOR(E,11,1):: NE
XT E :: FOR T=5 TO 8 :: CALL COL
OR(T,14,1):: NEXT T
9050 CALL COLOR(2,11,1,8,14,1,9,
PA,2,10,2,PA)
9060 RETURN
10000 REM ** HYPERSPACE **
10010 CALL SOUND(-4250,3000,25,2
000,25,1000,25)
10020 FOR E=1 TO 5 :: FOR T=1 TO
15 :: GOSUB 10400 :: CALL COLOR
(#T,CD):: NEXT T :: NEXT E :: CA
LL DELSPRITE(ALL)
10090 RETURN
10400 ON E GOTO 10410,10420,1043
0,10440,10450
10410 CD=13 :: RETURN
10420 CD=7 :: RETURN
10430 CD=14 :: RETURN
10440 CD=5 :: RETURN
10450 CD=1 :: RETURN
14999 REM ** MISSION COMPLETED *
*
15000 CALL DELSPRITE(ALL):: CALL
CLEAR :: DISPLAY AT(4,11):"
FANTASTIC
-----";
15010 DISPLAY AT(8,1):" YOU COMP
LETED THE MISSION -----"
15020 DISPLAY AT(14,2):"I BET YO
U CAN'T DO IT AGAIN"
15030 DISPLAY AT(23,3):"WERE WE
GO FOR ANOTHER PLAY"
15031 FOR N=110 TO 320 STEP 30 :
: CALL SOUND(250,N,0):: CALL SOU
ND(250,N,0,N+30,0):: CALL SOUND(
250,N,0,N+30,0,N+60,0):: NEXT N
15040 FOR T=1 TO 500 :: NEXT T :
: GOTO 352

```

TRENCH ATTACK

Extended Basic

IF YOU HAVE A PROGRAM YOU WOULD
LIKE TO SUBMIT FOR LISTING IN
KEYBOARD KRUNCHERS, OR IN FACT
ANY ARTICLE FOR THE MAGAZINE,
WRITE TO ME:

THE EDITOR,
99/4 MAGAZINE,
1 MANOR CLOSE,
WESTON,
HONITON, DEVON.
EX14 0PE

ALL LISTINGS ARE NOW AVAILABLE ON
TAPE AT A COST OF £2.50 EACH,
THIS INCLUDES POST & PACKING.
ORDERS TO THE EDITOR AS ABOVE.

NEXT MAG LOOK OUT FOR



THE FROG

```

10 REM Trench Attack II 22/6/86
20 REM by Peter Hutchison(c)
30 HISC=0 :: RANDOMIZE
40 SC,HITS=0 :: CALL MAGNIFY(4)
60 CALL SCREEN(2):: CALL CLEAR :
: BLK$=RPT$("F",16)
70 CALL CHAR(96,BLK$,97,"80C0E0F
0F8FCFEFF",98,"0103070F1F3F7FFF"
,99,"FF7F3F1F0F070301",100,"FFFE
FCF8F0E0C080")
80 CALL CHARPAT(97,A$,98,B$):: C
ALL CHAR(104,BLK$,105,A$,106,B$)
90 CALL COLOR(9,3,1,10,4,3,3,16,
3,4,16,3)
100 RESTORE :: FOR A=1 TO 30 ::
READ ROW,COL,CHR,ND :: CALL HCHAR
R(ROW,COL,CHR,ND):: NEXT A
110 DATA 1,1,96,32,2,1,96,2,2,3,
100,1,2,30,99,1,2,31,96,2,3,1,96
,1,3,2,100,1,3,31,99,1,3,32
120 DATA 96,1,4,1,100,1,4,32,99,
1,19,1,97,1,19,32,98,1,20,1,96,1
,20,2,97,1,20,31,98,1,20,32,96
130 DATA 1,21,1,96,2,21,3,97,1,2
1,30,98,1,21,31,96,2,22,1,96,96,
22,16,106,1,22,17,105,1,23,15,10
6
140 DATA 1,23,16,104,2,23,18,105
,1,24,14,106,1,24,15,104,4,24,19
,105,1
150 CALL COLOR(5,16,1,6,16,1,7,1
6,1,8,16,1)
160 DISPLAY AT(8,7):"TRENCH ATTA
CK II" :: DISPLAY AT(10,9):"BY A
CE GAMES"
170 DISPLAY AT(14,6):"PRESS FIRE
TO PLAY"
180 CALL KEY(1,FIRE,S):: IF FIRE
<>18 THEN 180
190 CALL HCHAR(8,1,32,7*32)
200 REM display stars
210 CALL CHAR(33,"0000001818")::
CALL COLOR(1,16,1)
220 FOR A=1 TO 60 :: CALL HCHAR(
INT(20*RND+2),INT(26*RND+4),33):
: NEXT A
230 REM set up sprites
240 CALL CHAR(108,"0000307878330
70F0F07337878300000000000C1E1ECCE
0F0F0E0CC1E1E0C")
250 CALL CHAR(112,"00200A002A841
04209601500410004802A80440928041
14401248042004")
260 CALL CHAR(116,"8040201008040
00000000408102040800102040810200
0000000201008040201")
270 CALL CHAR(120,"00040408111A1
E1B11080804040000002020101088587
8D8881010202")
280 CALL CHAR(124,"0000102828454
640212113130C0201000000040A0AD13
1814242E4E49820C")

```

```

290 CALL CHAR(128,"000040E050090
71C3F1F070950E040000000040E1420C
070F8F0C020140E04")
300 REM * Planet Approach *
310 FGT=INT(8*RND+5):: CALL DELSPRITE(ALL):: CALL SPRITE(#1,116,
12,88,112)
320 CALL SPRITE(#2,120+INT(3*RND
)*4,3,INT(97*RND+40),INT(193*RND
+32))
330 CALL JOYST(1,X,Y):: CALL KEY
(1,FIRE,S)
340 CALL MOTION(#1,SGN(-Y)*15,SG
N(X)*15)
350 IF FIRE<>18 THEN 430
355 FOR A=1 TO 4 :: CALL SOUND(1
00,-5,0):: NEXT A
360 CALL COLOR(#1,9):: CALL COIN
C(#1,#2,16,C):: IF C=0 THEN 410
370 CALL PATTERN(#2,112):: CALL
COLOR(#2,9):: CALL SOUND(600,-7,
0,110,0)
380 FGT=FGT-1 :: SC=SC+50 :: DIS
PLAY AT(23,8)SIZE(LEN(STR$(SC)))
:STR$(SC)
390 IF SC>HISC THEN DISPLAY AT(2
3,21)SIZE(LEN(STR$(SC))):STR$(SC
):: HISC=SC
400 CALL DELSPRITE(#2):: CALL CO
LOR(#1,12):: IF FGT=0 THEN 530 E
LSE 320
410 CALL COINC(#1,#3,16,C):: IF
C=0 THEN 430
420 SC=SC+10 :: CALL SOUND(600,-
7,0,110,0):: CALL DELSPRITE(#3)
430 CALL MOTION(#2,INT(2*RND)*7,
(2-INT(3*RND+1))*15):: CALL COLO
R(#1,12)
440 CALL POSITION(#2,ROW,COL)::
IF ROW<12 THEN CALL MOTION(#2,8,
0)ELSE IF ROW>152 THEN CALL MOTI
ON(#2,-8,0)
450 IF RND>.15 OR T>0 THEN 470
460 CALL POSITION(#2,ROW,COL)::
CALL SPRITE(#3,108,10,ROW,COL,7*
(ROW>96)-7*(ROW<96),7*(COL>128)-
7*(COL<128)):: T=8
470 CALL POSITION(#3,ROW,COL)::
T=T-1 :: IF T=0 AND ROW+COL>0 TH
EN CALL DELSPRITE(#3):: GOSUB 49
0 :: IF HITS=8 THEN 1160
480 GOTO 330
490 HITS=HITS+1 :: CALL SOUND(48
0,-7,0,700,15)
495 IF HITS/2<>INT(HITS/2) THEN R
ETURN
500 CALL COLOR(10,ABS(13*(HITS=2
)+11*(HITS=4)+10*(HITS=6)+14*(HI
TS=8)),3)
510 RETURN

```

```

520 REM set up trench and sprite
530 CALL DELSPRITE(ALL)
540 CALL CHAR(120,"80472B1122446
870684422112B67800002C4A81088442
C1C2C448810A8C402")
550 CALL CHARPAT(97,A$,98,B$)
570 CALL CHAR(65,BLK$,66,A$,67,B
$,88,BLK$,89,A$,90,B$,40,BLK$,41
,A$,42,B$,136,A$,137,B$)
580 CALL CHAR(72,BLK$,73,RPT$("0
",13)&"FFF",74,RPT$("0",9)&"FFFF
FFF",75,"00000"&SEG$(BLK$,1,11),
76,"0"&SEG$(BLK$,1,15))
590 CALL CHAR(77,"F0"&SEG$(BLK$,
1,14),78,"0000F0"&SEG$(BLK$,1,10
),79,"00000000F0FFFFFF",80,RPT$(
"0",12)&"FOFF")
600 CALL COLOR(5,5,13,6,13,5,7,1
3,5,8,9,13,2,5,13,14,3,13):: ROW
=7
610 A$=RPT$("A",11)&"IJKL MNOP"
&RPT$("A",11):: GOSUB 670 :: A$=
"AAAAAAIJKLHHHH HHHMNDOPAAAAA
A" :: GOSUB 670
615 A$="AAIJKLHHHHHHH* >HHHHHH
HMNDOPAA" :: GOSUB 670 :: A$="JK
L"&RPT$("H",10)&"CAAAAB"&RPT$("H
",10)&"MND"
620 GOSUB 670 :: A$=RPT$("H",12)
&"ZXXXXXXY"&RPT$("H",12):: GOSUB
670 :: A$=RPT$("H",11)&"CAAAAA
AAB"&RPT$("H",11)
625 GOSUB 670 :: A$=RPT$("H",10)
&"C"&RPT$("A",10)&"B"&RPT$("H",1
0):: GOSUB 670 :: A$=RPT$("H",9)
&"*"&RPT$("(",12)&"")&RPT$("H",9
)
630 GOSUB 670 :: A$="HHHHHHHH*"&
RPT$("(",14)&"")&HHHHHHHH :: GOSU
B 670 :: A$="HHHHHHHC"&RPT$("A",
16)&"BHHHHHHH"
635 GOSUB 670 :: A$="HHHHHHHC"&RP
T$("A",18)&"BHHHHHH" :: GOSUB 67
0 :: A$="HHHHHC"&RPT$("A",20)&"B
HHHHH" :: GOSUB 670
640 A$="HHHC"&RPT$("A",22)&"BHH
H" :: GOSUB 670 :: A$="H" & "HZ"&RP
T$("X",24)&"YH"
650 DISPLAY AT(21,1)SIZE(28):"H"
&RPT$("X",26)&"H" :: GOTO 700
660 REM Print Subroutine
670 FOR A=1 TO 32 :: CALL HCHAR(
ROW,A,ASC(SEG$(A$,A,1)):: NEXT
A
680 ROW=ROW+1 :: RETURN
690 REM * Planet Surface *
700 SCR=0 :: MINES=INT(16*RND+5)
:: CALL SPRITE(#1,116,12,88,112)
710 CALL JOYST(1,X,Y):: CALL KEY
(1,FIRE,S)

```

```

720 CALL MOTION(#1,SGN(-Y)*11,SG
N(X)*11)
730 IF FIRE<>18 THEN 800
740 CALL COINC(#1,#2,8,C):: CALL
COLOR(#1,9):: IF C=0 THEN 780
750 CALL PATTERN(#2,112):: CALL
SOUND(-485,-6,0,140,10):: SC=SC+
50 :: MINES=MINES-1
760 DISPLAY AT(23,8)SIZE(LEN(STR
$(SC))):STR$(SC):: CALL DELSPRIT
E(#2):: CALL COLOR(#1,12)
770 IF MINES=0 THEN 880 ELSE 800
780 CALL COINC(#1,#3,16,C):: IF
C=0 THEN 800
790 CALL DELSPRITE(#3):: SC=SC+1
0 :: DISPLAY AT(23,8)SIZE(LEN(STR
$(SC))):STR$(SC)
800 IF SC>HISC THEN HISC=SC :: D
ISPLAY AT(23,21)SIZE(LEN(STR$(SC
))):STR$(SC)
810 CALL POSITION(#2,ROW,COL)::
IF RND>.7 AND ROW+COL=0 THEN CAL
L SPRITE(#2,120,16,76,120,15,(2-
INT(3*RND+1))*8)
820 CALL POSITION(#3,ROW,COL)::
IF RND>.5 AND ROW+COL=0 THEN CAL
L SPRITE(#3,108,9,76,120,8,(2-IN
T(3*RND))*5)
830 CALL POSITION(#2,ROW,COL)::
IF ROW>136 AND ROW+COL>0 THEN CA
LL DELSPRITE(#2):: GOSUB 490 ::
IF HITS=8 THEN 1160
840 CALL POSITION(#3,ROW,COL)::
IF ROW>130 AND ROW+COL>0 THEN CA
LL DELSPRITE(#3):: GOSUB 490 ::
IF HITS=8 THEN 1160
850 IF SCR=0 THEN SCR=1 :: CALL
COLOR(2,9,13,8,5,13)ELSE SCR=0 :
: CALL COLOR(8,9,13,2,5,13)
860 GOTO 710
870 REM * Generator *
880 CALL DELSPRITE(ALL):: G1,G2=
0 :: T=100
890 CALL CHAR(120,"01020505091B3
36363331B09050502018040A09090D8C
CC6C6CCD89090A04080")
900 CALL SPRITE(#2,120,9,64,80,#
3,120,9,64,144,#1,116,12,130,110
)
910 CALL JOYST(1,X,Y):: CALL KEY
(1,FIRE,S)
920 CALL MOTION(#1,SGN(-Y)*10,SG
N(X)*10)
930 IF FIRE<>18 THEN 970
940 CALL COINC(#1,#2,16,C):: IF
C THEN CALL SOUND(100,-6,0):: CA
LL COLOR(#2,15):: G1=1
950 CALL COINC(#1,#3,16,C):: IF
C THEN CALL SOUND(100,-6,0):: CA
LL COLOR(#3,15):: G2=1
960 IF G1+G2=2 THEN SC=SC+1000 :
: GOTO 990

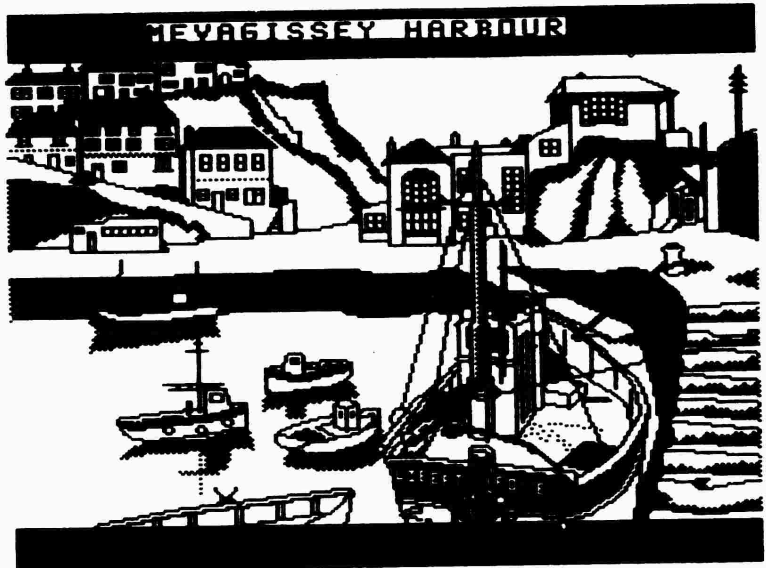
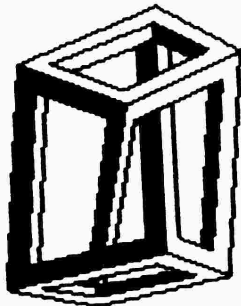
```

```

970 T=T-1 :: IF T>0 THEN 910 ELS
E CALL DELSPRITE(ALL):: GOTO 700
980 REM * Planet Destroyed *
990 CALL DELSPRITE(ALL):: FOR RO
W=7 TO 18 :: CALL HCHAR(ROW,1,32
,32):: NEXT ROW
1000 CALL HCHAR(19,2,32,30):: CA
LL HCHAR(20,3,32,28):: CALL HCHA
R(21,4,32,26)
1010 CALL CHAR(33,"0000001818"):
: CALL COLOR(1,16,1):: FOR A=1 T
O 60 :: CALL HCHAR(INT(20*RND+2)
,INT(26*RND+4),33):: NEXT A
1020 DISPLAY AT(23,8)SIZE(LEN(STR
$(SC))):STR$(SC)
1030 IF SC>HISC THEN HISC=SC ::
DISPLAY AT(23,21)SIZE(LEN(STR$(SC
))):STR$(SC)
1040 CALL CHAR(120,"030F1F3B755F
FFFFEBF77F6F3A1D0F03C070F8FCFEBE
F7EBF5FFBEFEFCF8F0C")
1050 CALL CHAR(124,"0000172D1054
2028207014310C1304000040D034881A
24150C041A4834C0308")
1060 CALL CHAR(128,RPT$("0",13)&
"1010303"&RPT$("0",24)&"80E0C08"
,132,RPT$("0",15)&"101"&RPT$("0"
,28)&"808")
1070 CALL SPRITE(#1,120,13,72,96
)
1080 FOR DELAY=1 TO 600 :: NEXT
DELAY
1090 CALL PATTERN(#1,124):: CALL
SOUND(1000,-7,0,15000,0)
1100 FOR DELAY=1 TO 100 :: NEXT
DELAY
1110 CALL DELSPRITE(#1)
1120 FOR DEBRIS=1 TO 14 :: CALL
SPRITE(#DEBRIS,128+4*INT(2*RND),
13,72,96,(2-INT(3*RND+1))*INT(11
*RND+5),(2-INT(3*RND+1))*INT(11*
RND+5))
1130 NEXT DEBRIS :: FOR DELAY=1
TO 1000 :: NEXT DELAY
1140 CALL DELSPRITE(ALL):: GOTO
240
1150 REM * End of Game *
1160 CALL DELSPRITE(ALL):: CALL
CLEAR :: CALL SCREEN(9):: CALL C
HARSET
1170 DISPLAY AT(8,7):"G A M E O
V E R"
1180 DISPLAY AT(11,5):"YOUR SCOR
E IS ";SC
1190 IF SC>HISC THEN HISC=SC ::
DISPLAY AT(14,6):"A NEW HIGH SC
ORE!"
1200 DISPLAY AT(21,1):"PRESS FIR
E FOR ANOTHER GAME"
1210 CALL KEY(1,FIRE,S):: IF FIR
E<>18 THEN 1210
1220 GOTO 40

```

T.I.T.BITS



YES.....RICHARD TWYNING STRIKES AGAIN !!!!!!!!!!!!!!!!!!!!!!!
 MEVAGISSEY HARBOUR, DRAWN USING GRAPHX, IS STUNNING
 THE IMPOSSIBLE CUBE
 DIRECTLY ABOVE YET
 ANOTHER STUNNER
 FROM RICHARD

ADD A TREMOLO TO YOUR SOUND
 WITH THIS SHORT ROUTINE
 VARIATION CAN BE OBTAINED
 BY CHANGING LINE 140 TO
 READ 1.01 or 1.03

```
100 FOR J=1 TO 60 STEP 2
110 READ A,B
120 FOR L=1 TO A
130 CALL SOUND(-1000,B,0)
140 CALL SOUND(-1000,B*1.02,0)
150 NEXT L
160 NEXT J
170 CALL SOUND(-1,300000,30)
180 DATA 2,330,2,294,4,330,4,294
    ,4,330,4,294,4,262,8,220
190 DATA 2,330,2,294,6,330,2,294
    ,4,330,4,262,12,247
200 DATA 2,294,2,262,4,294,4,262
    ,4,294,2,330,2,294,4,262,8,220
210 DATA 4,262,4,220,4,262
    ,4,247,16,220
```

PROFESSIONAL LOOK BORDER

 TRY THIS AT THE BEGINNING
 OF YOUR PROGRAM.BUT DO'NT
 USE COLUM 1 or 28 IN YOUR
 PROGRAM

```
100 CALL CLEAR
110 CALL SCREEN(5)
120 CALL VCHAR(1,31,1,96)
130 FOR SET=1 TO 12
140 CALL COLOR(SET,2,16)
150 NEXT SET
160 GOTO 160
```

TO PASS FROM EXT.BASIC
 TO BASIC TYPE THIS :-

```
1 !@P-
2 CALL COLOR
```

GRAPHX OFFERS THESE
 "COMPUTER ANIMATED"
 FEATURES: USE YOUR
 IMAGINATION TO CREATE
 ACTION GRAPHICS!!!

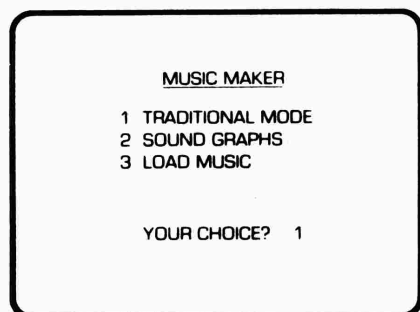


REVIEW
REVIEW
REVIEW

Music Maker

REVIEW
REVIEW
REVIEW

AFTER BEING TREATED TO A CLASSICAL PIECE (THE WRITER'S NAME ESCAPES ME), YOU ARE GIVEN A MENU:-



OPTION 1 ALLOWS YOU TO COMPOSE A TUNE USING CROTCHETS, QUAVERS, ETC.

OPTION 2 ALLOWS YOU TO COMPOSE, WHETHER YOU CAN READ/WRITE MUSIC OR NOT. THIS PARTICULAR OPTION WAS WRITTEN IN SPECIFICALLY FOR SOUND EFFECTS FOR GAMES. YOU CAN ALSO COMPOSE USING THE 8 NOISE CHANNELS, CHANGE NOTE VALUES, ETC.

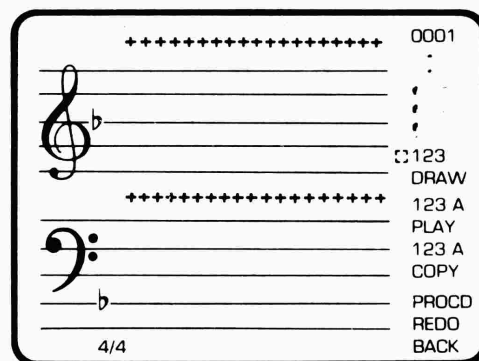
OPTION 3 ALLOWS YOU TO LOAD PREVIOUSLY SAVED MUSIC FROM DISK OR CASSETTE.

THE PIECE DE RESISTANCE IS UNDOUBTEDLY THE "TRADITIONAL MODE". LIKE MOST UTILITY PROGRAMS IT HAS IT'S FAULTS WHICH WILL BE LISTED LATER.

AFTER SELECTING OPTION 1 YOU ARE ASKED "NUMBER OF SHARPS" - DEFAULT BEING ZERO - THIS ENABLES YOU TO WORK WITH A KEY SIGNATURE. SHOULD YOU PRESS ENTER YOU WILL BE ASKED "NUMBER OF FLATS". THE DEFAULT AGAIN BEING ZERO. IF YOU SELECTED NO SHARPS OR FLATS THEN YOU WILL BE COMPOSING IN THE KEY OF C/A MINOR. THE NEXT QUESTION "TIME SIGNATURE" - DEFAULT 4/4. THERE ARE 6 SIGNATURES TO CHOOSE FROM:- 2/2, 2/4, 3/4, 4/4, 6/8 AND 8/16. 3/4 = WALTZ, 4/4 = COMMON TIME, ETC. AFTER THAT YOU CAN CHOOSE THE TEMPO FROM 1 TO 30 - DEFAULT 15. 1 IS REALLY SLOW (SNAILS PACE) TO 30 (ALLEGRO).



NOW WORK CAN BEGIN. THE TRADITIONAL MODE MEASURE APPEARS. A CYAN SCREEN WITH GREYISH MANUSCRIPT AND BLUE LINES, BLACK TREBLE AND BASS CLEFS. THE NOTES ARE INITIALLY BLACK UNTIL YOU CHOOSE FROM THE 3 VOICES MARKED DRAW 1,2,3 THESE ARE COLOURED BLUE=DRAW 1. RED=DRAW 2 AND GREEN=DRAW 3. ALL OPTIONS CAN BE ACCESSED USING EITHER THE KEYS OR JOYSTICK 2. I USE BLUE FOR MELODY, RED FOR BASS AND GREEN FOR COUNTER MELODIES AND CHORDS. YOU CAN PLAY ALL THE VOICES AT ONCE OR INDIVIDUALLY. YOU HAVE A SELECTION OF RESTS ALSO. THE COPY OPTION IS A GEM.



Music Maker



IF YOU WRITE A MEASURE/BAR AND THE SECOND MEASURE IS THE SAME, OR PART OF THE SECOND MEASURE IS THE SAME, THEN CHOOSE EITHER 1, 2, 3 OR ALL AND THE BAR IS COPIED, SAVING YOU TIME AND ALLOWING YOU TO GET ON WITH THE NEXT PART OF YOUR TUNE. SHOULD YOU NOT LIKE ANYTHING, JUST USE THE ERASE OPTION WHICH IMMEDIATELY APPEARS OVER THE COPY OPTION AS SOON AS A NOTE IS PLACED ON ANY OF THE STAVES. YOU STILL HAVE 3 OTHER OPTIONS TO DEAL WITH WHICH ARE, PROC'D, REDO OR BACK.

PROC'D	-	PROCEEDS TO THE NEXT MEASURE
REDO	-	TAKES YOU TO THE PREVIOUS MEASURE
BACK	-	TAKES YOU TO ANOTHER MENU, 4 OPTIONS ARE NOW AVAILABLE:

- | | |
|---------|----------|
| 1. EDIT | 3. SAVE |
| 2. PLAY | 4. PRINT |

OPTION 1 - LETS YOU EDIT THE TEMPO AND ALSO ALLOWS YOU TO GET INTO ANY SPECIFIED MEASURE.

OPTION 2 - LETS YOU PLAY ALL OR PART OF YOUR COMPOSITION AND AGAIN ALLOWS YOU TO CHANGE THE TEMPO. TRANSPOSING UP AND DOWN IS POSSIBLE.

OPTION 3 - LETS YOU SAVE TO DISK OR CASSETTE.

OPTION 4 - LETS YOU OBTAIN A PRINT OUT IF YOU HAVE A PRINTER.

FCTN 5 IF PRESSED TELLS YOU "ALL MUSIC IS LOST FROM THIS POINT. ARE YOU SURE Y/N" : N=DEFAULT. PRESSING Y AND ENTER WIPES EVERYTHING OUT, SO BE CAREFUL. THE MAIN MENU IS DISPLAYED AND YOU CAN START AGAIN.

AS I SAID, THE MODULE HAS A FEW FAULTS, AND THEY ARE:-

- 1) NO CHANGING KEY AND TIME SIGNATURE.
- 2) IF YOU HAVE THE NOTE "A" 3 TIMES IN A MEASURE BUT YOU WANT IT TO BE "A SHARP" THEN YOU HAVE TO CHANGE THOSE 3 NOTES TO "A SHARP".
- 3) TRIPLETS ARE NOT CATERED FOR.
- 4) DEMI SEMI QUAVERS ARE NOT CATERED FOR.
- 5) ONLY THREE OCTAVES ARE AVAILABLE A TO A.
- 6) YOU ONLY GET A MINIMAL AMOUNT OF ACCIDENTALS.

THE MODULE'S GOOD POINTS ARE ENDLESS AND IT IS A VERY GOOD BUY.

HELP

BY MIKE CURTIS



VOODOO CASTLE

THERE ARE 25 LOCATIONS, 61 VERBS, 87 NOUNS, 97 MESSAGES
AND 65 OBJECTS. YOU CAN ONLY CARRY 9 OBJECTS AT A TIME.

SOME VERBS ARE :-

GO	MOVE	BREAK	GET	DROP	DUST	DRINK	CLEAN
SAW	MIX	HAMMER	LISTEN	HEAR	DIG	ON	WAVE
HELP	PRESS	PUSH	SMASH	REACH	OPEN	CLOSE	TURN
LOOK	SAW	SHAKE	DANCE	YELL	SCREAM	SAY	CUT
STAB	RUB	HUG	*SMOKE	CIRCLE	READ	SUMMON	ENTER
KICK	CLIMB	THROW	SHRINK				

SOME NOUNS ARE :-

NORTH	SOUTH	EAST	WEST	UP	DOWN	KNIFE	COFFIN.
IDOL	DOOR	KETTLE	HOLE	SWORD	BOARD	PAPER	STICK
PAGE	GRAVE	CHEMICAL	FIRE	SAW	STAIR	SHOVEL	ROCK
WINDOW	CLOVER	CHUTE	LABEL	WALL	MEDIUM	CHIMNEY	BUTTON
SHIELD	SAFE	BREAK	ARMOUR	GLASS	RING	TUBE	FIRE
FLUE	SOOT	CHIMNEY	33 34	35 36	37 38	PIN	FOOT
SWEEP	MAN	DOLL	HAMMER	BOOK	ZAP	*POT	CELLAR
CRY	LEAF	WRITING	LEAFLET	BALL	NAIL	CRACK	AROUND
LOCK	LEDGE	VOICE	SIGN	FLOOR	CHANT	SOUP	LABEL
HEAD	BAG						

THATS GOT THE GENERAL INFORMATION OUT OF THE WAY, NOW FOR
SOME HELP!

THIS ADVENTURING REQUIRES YOU TO REMOVE THE CURSE FROM THE COUNT.
FAIRLY OBVIOUS YOU MIGHT SAY, BUT BEWARE OF CELL DOORS SLAMMING
SHUT ON YOU WHEN YOU NEED TO GO THERE, BREAKING CRYSTAL BALLS,
YOU ALSO HAVE TO DO A BIT OF SAFECRACKING TO HELP A CHIMNEY-SWEEP
OUT OF A JAM, AND DO A SPOT OF CHEMISTRY TO STUNT YOUR GROWTH.
YOU WILL FIND IT NECESSARY TO YELL A COUPLE OF TIMES, THE BAG HOLDS
QUITE A LOT DOESN'T IT, TRY WAVING IT AROUND AS WELL. FOR SMALL
PRINT, GLASS MAGNIFIES!, A BROKEN SWORD IS STILL USEFUL, IT GETS
THINGS OFF WALLS AS WELL, I CAN RECOMMEND THE SOUP BUT THE BREW
IS AWFUL.

THIS IS NOT A TERRIBLY HARD ADVENTURE ON REFLECTION, BUT IT
HAS GOT A FEW TWISTS, AND MORE THAN ONE WAY TO GET KILLED OFF!
AND YOU ALSO GET WHISKED FROM ONE LOCATION TO ANOTHER, BY LOOKING
IN A CRYSTAL BALL, DROPPING SOOT ON THE FLOOR, AND THE ODD MAGIC
WORD OR TWO COMES IN HANDY AS WELL.
TRY THE WORDS MARKED WITH AN ASTERISK NEAR THE WITCHES BREW AND
DON'T GET CAUGHT.

PIRATE ADVENTURE.

THERE ARE 26 LOCATIONS , 58 VERBS , 76 NOUNS , 88 MESSAGES AND 64 OBJECTS. YOU CAN ONLY CARRY 6 OBJECTS AT A TIME.

THE VERBS ARE :- WEAR, PULL, FLY, EMPTY, WEIGH, SAIL, WAKE, GO/CLIMB/WALK/RUN/PACE/ENTER/FOLLOW, SAY, UNLIGHT, HELP, SWIM, LOOK/EXAMINE/WATCH, READ, LISTEN, KILL/ATTACK, BURY, FIND, JUMP, GET/TAKE/CATCH/PICK, BREAK/SMASH, SET, CAST, DIG, WAIT, FEEL, DRINK/EAT, DROP/THROW/LEAVE/GIVE/RELEASE, LIGHT, OPEN, SHAKE, BUILD/MAKE, CLOSE/SHUT

THE NOUNS ARE :- NORTH, SOUTH, EAST, WEST, UP, DOWN, PASSAGE BOTTLE/RUM, MONGOOSE, BAG/DUFFEL, PIRATE, TORCH, OFF, MATCH, YOHO, LUMBER, RUG, KEY, AROUND, DOUBLOON, SAIL, FISH, ANCHOR, SHACK, PLACE, CAVE, SIGN, DOOR, CHEST, PARROT, HAMMER, NAILS, BOAT/SHIP, SHED, WATER/SALT, LAGOON/TIDE, PIT, MAP, PACE, 30, SHORE/BEACH, SAND, HOLE, BOX, SNEAKERS, CRACKERS/SACK, PIECE, KEEL, FLOTSAM/JETSAM, STAMPS/UPS?, PATH/HILL, NOTE, HALL, DESTRUCTIVE, CROCODILE, SNAKE, TREASURE, FLY, WINDOW, BOOK

THE OBJECT OF THIS ADVENTURE IS TO RETREIVE THE TREASURES IN A FLAT IN LONDON? THIS ADVENTURE IS PERFECT TO BEGIN WITH AS IT FEATURES ALMOST ALL ASPECTS OF ADVENTURING THAT YOU ARE LIKELY TO COME UP AGAINST IN A SCALED DOWN VERSION.

1. EXCHANGING LOCATIONS BY A MAGIC WORD (YOHO)
2. A COLLECTION OF ITEMS TO MAKE ANOTHER (SAILS, WOOD ETC.=BOAT)
3. ITEM NOT VISIBLE UNTIL ANOTHER PRESENT (OPEN CHEST=PLANS)
4. TWO ITEMS TOGETHER AND THEY BOTH VANISH (FISH & CROCS)
5. HELPFUL ADVICE FROM ANOTHER ITEM (PARROT)
6. TIME LIMIT IMPOSED BY TORCH RUNNING OUT OF FUEL
7. MESSAGES FROM WITHIN TO ADVISE AND GUIDE YOU, IF IT SAYS 'I DON'T KNOW HOW TO -' THEN TRY ANOTHER VERB. IF IT SAYS 'I DON'T KNOW WHAT A - IS' THEN TRY ANOTHER NOUN. IF IT SAYS 'I CAN'T DO THAT YET' THEN AN ITEM IS NOT PRESENT THAT WILL COMPLETE THE COMMAND OR YOU ARE NOT AT THE RIGHT LOCATION YET.
8. HIDDEN EXITS REVEALED BY MOVING ITEMS, UNLOCKING THINGS OR EVEN BREAKING THEM.

THATS GOT THE GENERAL INFORMATION OUT OF THE WAY, NOW FOR SOME HELP! THIS IS A FAIRLY EASY START TO ADVENTURING AND WILL GIVE YOU A LOT OF INSIGHT OF WHAT IS IN STORE WITH THE REST OF THE SERIES OF PROGRAMS, SO TOO MUCH HELP WILL NOT HELP REALLY, SO NOW ALL YOU HAVE TO DO IS WORK OUT WHAT, WHEN AND WHERE! REMEMBER, LOOK AT IT, MOVE IT, FEEL IT, GET IT, READ IT, DESCRIBE IT, BREAK IT, WEAR IT, DIG IT, THROW IT, BE ADVENTUROUS - THAT'S WHAT THESE PROGRAMS ARE FOR!!

A FEW TIPS :- YOU CAN SAY 'YOHO' ANYWHERE, AS LONG AS YOU HAVE THE BOOK! THERE IS A LOCKED DOOR, YOU NEED SOMETHING TO FASTEN PLANKS TOGETHER WITH? I WONDER WHAT HOLDS CARPETS DOWN? SMALL BIRDS EAT WORMS, WHAT DO BIG BIRDS EAT APART FROM CRACKERS? OR WHAT DO CROCODILES EAT? CAN YOU SWIM? WEAR THE SHOES! WHAT DO YOU USE TO SWIM? DO NOT LOSE THE SHOVEL, OR LEAVE IT BEHIND WHEN YOU SET SAIL. BEHIND, SOME TREASURES ARE USUALLY BURIED. TURN THE TORCH OFF WHEN NOT NEEDED AS IT CANNOT BE RE-FUELED IN THIS ADVENTURE. THE PIRATE ALSO GETS THIRSTY AFTER A LONG VOYAGE, SO YOU WILL HAVE TO FIND SOMETHING TO QUENCH HIS THIRST FOR THE RETURN TRIP, OR HE WILL LEAVE YOU STRANDED ON THE ISLAND, REMEMBER YOU CANNOT TAKE THE BOOK ON THE SHIP, SO 'YOHO' IS NOT GOING TO WORK ON TREASURE ISLAND. IF I GIVE TOO MANY CLUES YOU WILL HAVE NOTHING LEFT TO DO. THE LISTED VERBS AND NOUNS SEPARATED WITH A '/' ARE SYNONYMS.

- A LITTLE QUIRK, TRY TYPING "WALK KQF" WHEN ON TREASURE ISLAND. -

BRIGHT SPARKS



Should this article be in the Sound Advice column or the column for Bright Sparks, probably both since it is all about using sound in assembly language. Working out the correct values of frequency, duration and volume for sound seems hard enough in basic, but if any of you have read the chapter on sound in the EDITOR/ASSEMBLER MANUAL will realise easy the basic method really is (but time consuming). In this tutorial I will hope to show you how to work out a sound table (using the short extended basic program) and how to incorporate them into your own assembly language programs, if you own either the minimemory or the E/A that is.

Like graphics you must have a table in the VDP ram which contains the relevant values, but unlike graphics there is only one table which contains all the information about a particular sound. However it is not as simple as just stating the duration, frequency and volume as I will explain.

The first byte of a sound list as they are called must contain the number of bytes in the list excluding the duration. Each sound list must contain the frequency bytes or the volume byte or both for 1 to 3 of the sound channels and lastly the volume byte. That is a bit of a mouthful so here is a table showing how a typical sound list is made up.

```

*****
*   BYTE   *   BITS   *   CONTENTS   *
*****
*   *   *   *   *
*   1   *   0-7   *   THE NUMBER OF BYTES IN THE SOUND *
*   *   *   *   * LIST EXCLUDING THE DURATION *
*****
*   *   *   *   *
*   2   *   *   *   * FREQUENCY BYTE 1 *
*   *   *   0   *   1 *
*   *   *   1-3 *   000,010 OR 100 DEPENDING IF YOU *
*   *   *   *   * ARE USING CHANNEL 1,2 OR 3 *
*   *   *   4-7 *   4 LSB OF THE FREQUENCY CODE *
*****
*   *   *   *   *
*   3   *   *   *   * THE SECOND FREQUENCY BYTE *
*   *   *   0-1 *   00 *
*   *   *   2-7 *   6 MSB OF THE FREQUENCY CODE *
*****
*   *   *   *   *
*   4   *   *   *   * BYTE FOR THE VOLUME *
*   *   *   0   *   1 *
*   *   *   1-3 *   001,011 OR 101 DEPENDING IF YOU *
*   *   *   *   * ARE USING VOICE 1,2 OR 3 *
*   *   *   4-7 *   THE VALUE FOR THE VOLUME *
*   *   *   *   * 0000 LOUDEST TO 1111 NO SOUND *
*****
*   *   *   *   *
*   5   *   0-7   *   DURATION >00 NO DURATION TO *
*   *   *   *   * >FF WHICH IS JUST OVER 4 SECS. *
*****

```

There are one or two interesting things about the table. maybe you noticed that there are only 10 bits to specify the frequency, this means that the computer can only play a maximum of 1024 (2^{10}) different tones since the maximum different binary combinations is 1024. The actual value you use for the sound lists is not the actual frequency but the frequency code, which is calculated by the formula:-

FREQUENCY CODE=111860.8/FREQUENCY

The other is that there are only 4 bits for the volume giving 15 different values whereas the basic manual would have you believe there are 30 different values.

It does not really matter if you have not understood all or any of this since I have written a small program which will calculate the actual assembler values for frequency, volume and duration from the values you would use in basic. (Allowing you to pinch other peoples sound effects from magazine listings.)

I think it would be best to show you how to use this program by using a few examples.

1. To convert the values of CALL SOUND (500,110,0) to a sound list.

Firstly run the program (After typing it in of course). When the channel prompt appears on the screen type "1" and press ENTER. When the duration prompt appears type "500" and press ENTER. Type "110" for frequency and then "0" for volume. After a brief pause the hexadecimal values for the sound should appear at the bottom of the screen. This is not quite the finished sound list since the first byte is not printed. For this sound list it is 3, since there are 2 bytes for the frequency and 1 for the volume, the duration byte is never included. So the completed sound list would be:-

```
NO.   FREQ  VOL  DUR
>03,>89,>3F,>91,>1E
```

To stop the above sound list you need another one, this can be to play another sound or to stop the sound altogether. To stop the sound you must set the duration to 0 and the volume to 32 (on the channel you want to stop, in this case 1). You should use the convertor program once again to find out the assembler values, but this time when the frequency prompt appears just press ENTER. The values appearing on the bottom of the screen should be >9F,>00 to give the sound list:-

```
>01,>9F,>00
```

2 If you want to play a chord say the equivalent to CALL SOUND (500,110,0,220,5)

Firstly you would do the same as above to get the values for (500,10,0). Now you must find the values for (500,220,5) making sure you specified sound channel 2. The number of

frequency and volume bytes is 6 so the first byte in the sound list would be 6. Below is the compleated sound list.

```
NO.    FREQ1 VOL1 FREQ2  VOL2 DUR
>06 >89 >3F >91 >AC >1F >B2 >1E
```

This time when you finish you must stop both sound channels by giving them both a volume of 32 and a duration of 0. Here is the sound list that will do this:-

```
NO.    VOL1 VOL2 DUR
>02 >9F >BF >00
```

Hopefully you will now be able to generate your own sound tables but obviously this is not much good without a program to execute them. Below is a program/subprogram which will play firstly a tone (110Hz) then a chord (110Hz & 220Hz) then stop the sound. The same program can be made to play any sound list with a few modifications as you will find out later.

EDITOR/ASSEMBLER VERSION

```
DEF SOUND1 *
REF VMBW * SECTION1
SNDBUF EQU >1000 *
H01 BYTE >01 *
EVEN *
SOUND1 *
LI R0,SNDBUF *
LI R1,SNDBL * SECTION2
LI R2,17 *
BLWP @VMBW *
LIMI 0 *
LI R10,BUFFER *
MOV R10,@>83CC * SECTION3
SO CB @H01,@>83FD *
MOVB @H01,@>83CE *
LIMI 2 *
LOOP MOVB @>83CE,@>83CE *
JNE LOOP * SECTION4
LIMI 0 *
RT *
SNDBL BYTE >03,>89,>3F,>91,>1E *SECTION5
BYTE >06,>89,>3F,>91,>AC,>1F,>B2,>1E
BYTE >02,>9F,>BF,>00
END
```

MINI-MEMORY

```
LI R0,>1000 *
LI R1,ST * S2
LI R2,17 *
BLWP @>6028 *
LIMI 0 *
LI R10,>1000 * S3
MOV R10,>83CC *
LI R10,>0100 *
SO CB R10,@>83FD *
MOVB R10,@>83CE *
LIMI 2 *
LP MOVB @>83CE,@>83CE *
JNE LP * S4
LIMI 0 *
B *R11 *
ST DATA >0389,>3F91,>1E06 *S
DATA >893F,>91AC,>1FB2 *
DATA >1E02,>9FBF,>0000 *
END
```

I will firstly deal with the editor/assembler listing. Firstly you should note that the parts of the program beginning with the asterisks are remarks and need not be typed in, secondly it should be assembled with the R option of the assembler.

As the remarks show I have divided the program into several sections, I shall now discuss these sections in some detail.

SECTION 1: The first thing it does is to specify the programs name using the DEF command, secondly tells the computer that the program will be using the VMBW routine. It also sets the word at SNDBUF to >1000 as this is where the sound table is going to be in the VDP ram.

SECTION 2: This is the routine which places the sound lists into the sound buffer starting at >1000 in the VDP ram. There is one thing that you must change here if you are using your own sound lists, R2 contains the total number of bytes in the sound table (which is contained in section 5 of the program). In this case it is 17 since in the three sound lists there are total of 17 bytes.

SECTION 3: You don't need to bother much about this bit as it will be the same no matter what sound lists you are wanting to play. This is the part of the program which does the sound processing.

SECTION 4: In this program it just waits until the sound list is finished and uses the RT command to return to the editor/assembler menu. Although if you are careful you can have the computer drawing a picture, or anything else for that matter while playing the sound table. To do this although if you are writing to VDP ram then you must dissable the interrupts, by using LIM1 0, or else the computer will crash. You must also enable the interrupts several times a second, using LIM1 2 to allow the sound to be played. The best idea is to do this in the main loop of your program.

LIMI 2
LIMI 0

If you do not want to do this you can just let the sound list finish before your program continues, only replacing RT by the next line of your program.

SECTION 5: This is the section where the sound lists are stored. The editor assembles BYTE directive makes things quite easy since each sound list may be stored on a separate line. I have used the example that I showed you earlier on so you could see how the sound lists are used in a program.

Instead of repeating all this for the mini memory version I will only discuss the differences between the two programs. The first difference is that there is no SECTION 1 since the program at present has no name, instead is run by the easy bug command E7D00, and to save memory the values of the buffers, etc. are loaded directly into registers rather than using labels. In SECTION2 VMBW is replaced by >6028 since this is the address of the VMBW routine in the mini memory. In SECTION5 RT is replaced by B *R11 since the line by line assembler does not have the RT instruction, they both do the same thing anyway. The biggest difference is that the mini memory does not have the BYTE directive so you must use the DATA directive instead. This is not as handy since each new sound list cannot always begin on a new line, making it slightly more difficult to edit a sound list. One advantage of using the mini memory is that it is easy to change the sound list by using easy bug without retyping the whole program.

To do this is quite simple, once you have checked that the program has worked and saved it on tape you can start. To change the sound table enter the easybug command M7D3A. You are now free to enter a new sound table a byte at a time, or just to change the current one. Once you have done this you press "." then type M7D0A and type the total number of bytes in the new sound table (this value must be in hexadecimal since easybug does not handle decimal numbers).

I hope this article helps you conquer the TMS9919 sound processor.

PS: If you want to see anything put in the magazine that is not already covered please write in, it could lead to a better magazine (if that were possible).

```

100 DIM H$(16)
110 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
120 FOR A=0 TO 15 :: READ H$(A):: NEXT A
130 GOTO 150
140 A$,B1,B1$,B2,B2$,B3,B3$,B4$,A1$,A2$,A,B,N,BIN$,CHAN,F$,FREQ,V$,VOL,DUR :: CALL
DECBIN :: CALL BINDEC :: CALL CLEAR :: !@P-
150 CALL CLEAR :: ON WARNING NEXT :: PRINT "FREQUENCY  VOLUME  DURATION  BYTES
    BYTE      BYTE  "
160 DISPLAY AT(1,1):"CHANNEL" :: DISPLAY AT(2,1):"DURATION" :: DISPLAY AT(3,1):"
FREQUENCY" :: DISPLAY AT(4,1):"VOLUME"
170 ACCEPT AT(1,9)VALIDATE("123")BEEP SIZE(1):CHAN :: ACCEPT AT(2,10)VALIDATE(DI
GIT)BEEP:DUR :: ACCEPT AT(3,11)VALIDATE(DIGIT)BEEP:F$
180 DUR=MIN(INT(DUR*.06+.03),255):: CALL DECHEX(DUR,B4$,H$())
190 ACCEPT AT(4,8)BEEP VALIDATE(DIGIT):V$
200 IF F$="" THEN 260 ELSE FREQ=MAX(VAL(F$),110)
210 FREQ=INT(111860.8/FREQ+.5):: CALL DECBIN(FREQ,A1$)
220 CALL DECBIN((CHAN-1)*2,A2$)
230 A1$=RPT$("0",10-LEN(A1$))&A1$ :: A2$=RPT$("0",3-LEN(A2$))&A2$
240 A$="1"&A2$&SEG$(A1$,7,4):: CALL BINDEC(A$,B1):: CALL DECHEX(B1,B1$,H$())
250 A$="00"&SEG$(A1$,1,6):: CALL BINDEC(A$,B2):: CALL DECHEX(B2,B2$,H$())
260 IF V$="" THEN 330 ELSE VOL=MIN(VAL(V$),30)
270 CALL DECBIN(INT(VOL/2),A1$)
280 A1$=RPT$("0",4-LEN(A1$))&A1$
290 CALL DECBIN((CHAN-1)*2+1,A2$):: A2$=RPT$("0",3-LEN(A2$))&A2$
300 A$="1"&A2$&A1$ :: CALL BINDEC(A$,B3):: CALL DECHEX(B3,B3$,H$())
310 IF F$="" THEN 320 ELSE PRINT TAB(2);">";B1$;",">";B2$;TAB(13);">";B3$;TAB(22)
; ">";B4$ :: GOTO 160
320 PRINT TAB(13);">";B3$;TAB(22);">";B4$ :: GOTO 160
330 PRINT TAB(2);">";B1$;",">";B2$;TAB(22);">";B4$ :: GOTO 160
340 !@P+
350 REM *CONVERT BINARY TO    DECIMAL*
360 SUB BINDEC(BIN$,A)
370 N=.5 :: A=0
380 FOR B=LEN(BIN$)TO 1 STEP -1 :: N=N*2 :: A=A+VAL(SEG$(BIN$,B,1))*N :: NEXT B
390 SUBEND
400 REM *CONVERTS DECIMAL TO BINARY*
410 SUB DECBIN(A,BIN$)
420 BIN$=""
430 IF A/2<>INT(A/2)THEN BIN$="1"&BIN$ ELSE BIN$="0"&BIN$
440 A=INT(A/2):: IF NOT A<1 THEN 430
450 SUBEND
460 SUB DECHEX(A,A$,H$())
465 A$=""
470 A$=H$((A/16-INT(A/16))*16)&A$
480 A=INT(A/16):: IF A>0 THEN 470
490 A$=RPT$("0",2-LEN(A$))&A$
500 SUBEND

```

KIDS CORNER

PAGE 29

"MOVE IT" BY N.WOLSLEY-CHARLES
A NICE SHORT PROGRAM WHICH IS SELF EXPLANATORY

PAGE 30

"MENTAL EXERCISE" BY M. MENDES

TWO PUZZLES IN THIS PROGRAM NO.1 THE CHOSEN ROW & COLUMN SQUARES CHANGE FROM RED TO BLUE & VICE VERSA. THE PUZZLE IS SOLVED WHEN THE 1st & 4th COLUMNS ARE RED & THE 2nd & 3rd ARE BLUE NO.2 THE CHOSEN ROW & COLUMN SQUARES WILL INCREASE BY 1 (THE SQUARE WITH A VALUE OF 5 WILL TURN TO 0) THE PUZZLE IS SOLVED WHEN ALL THE SQUARES ARE 0.

```
100 RANDOMIZE :: CALL CLEAR :: C
ALL SCREEN(15):: CALL COLOR(12,5
,15,11,14,15,2,7,15)
110 CALL CHAR(95,"000000FFFF")
120 CALL CHAR(112,"00000003070E1
C18")
130 CALL CHAR(113,"000000C0E0703
818")
140 CALL CHAR(114,"181C0E0703")
150 CALL CHAR(115,"183870E0C0")
160 CALL CHAR(116,"000000FFFF")
170 CALL CHAR(117,"1818181818181
818")
180 CALL CHAR(120,"C0C00C0CC0C00
C0C")
190 PRINT "          PARCO PRESENT
S          ":
200 PRINT "          >> M O V E   I T
<<
```

```
XXXXXXXXXXXXXXXXXXXXX"
210 PRINT "          xpta pta pta pt
ax          xu u u u u u u ux
          xrts rts rts rtsx"
220 PRINT "          x
          MOVES xpta pta pta ptax B
EST_____xu u u u u u u ux ____"
230 PRINT "          xrts rts rts rt
sx          x          x
          xpta pta pta ptax"
240 PRINT "          xu u u u u u u
ux          xrts rts rts rtsx
          x          x"
250 PRINT "          xpta pta pta
x          xu u u u u u * x
          xrts rts rts x"
260 PRINT "          XXXXXXXXXXXXXXXX
xx":
270 DX=23
280 DY=20
290 M=0
300 CALL HCHAR(24,1,32,32)
310 CALL HCHAR(14,1,32,7)
320 A=INT(RND*15)+1
330 IF A=1 THEN 320
340 D=A
350 FOR C=8 TO 20 STEP 4
360 FOR B=11 TO 23 STEP 4
370 CALL VCHAR(C,B,A+64)
```

```
380 A=A+1
390 IF A>16 THEN 410
400 A=1
410 IF A=D THEN 440
420 NEXT B
430 NEXT C
440 CALL JOYST(1,JX,JY)
450 NX=DX :: NY=DY
460 JZ=(JX+(JY*2)+12)/4
470 ON JZ+1 GOTO 440,480,490,440
,500,510,440
480 NY=DY-4 :: GOTO 520
490 NX=DX+4 :: GOTO 520
500 NX=DX-4 :: GOTO 520
510 NY=DY+4
520 IF (NX<11)OR(NX>23)OR(NY<8)O
R(NY>20)THEN 440
530 M=M+1 :: DISPLAY AT(14,1)SIZ
E(4):M
540 DISPLAY AT(DY-1,DX-3)SIZE(3)
:"pta" :: DISPLAY AT(DY,DX-3)SIZ
E(3):"u u" :: DISPLAY AT(DY+1,DX
-3)SIZE(3):"rts"
550 CALL GCHAR(NY,NX,CH)
560 CALL VCHAR(DY,DX,CH)
570 FOR A=NY-1 TO NY+1
580 CALL HCHAR(A,NX-1,32,3)
590 NEXT A
600 CALL VCHAR(NY,NX,42)
610 DX=NX :: DY=NY
620 CCH=0
630 FOR YCH=8 TO 20 STEP 4
640 FOR XCH=11 TO 23 STEP 4
650 CCH=CCH+1
660 IF CCH=16 THEN 710
670 CALL GCHAR(YCH,XCH,CH)
680 IF CH>CCH+64 THEN CALL SOUN
D(-100,1000,2):: GOTO 440
690 NEXT XCH
700 NEXT YCH
710 IF M>BM THEN BM=M
720 DISPLAY AT(14,25)SIZE(4):BM
730 DISPLAY AT(24,3):"ANOTHER GA
ME? UP=Y/DOWN=N"
740 CALL JOYST(1,JX,JY)
750 IF JY=0 THEN 740
760 IF JY=4 THEN 270
770 CALL CLEAR
780 END
```

Extended Basic



MENTAL EXERCISE

TI Basic

```

100 RANDOMIZE
110 CALL CLEAR
120 CALL SCREEN(5)
130 CALL CHAR(97,"FF818181818181
FF")
140 CALL CHAR(104,"FF818181818181
1FF")
150 CALL CHAR(143,"007E424242427
E00")
160 CALL CHAR(134,"FFFFFFFFFFFFF
FFF")
170 CALL COLOR(9,16,7)
180 CALL COLOR(10,16,5)
190 PRINT "      MENTAL EXERCISE
"
200 PRINT :::::::::::
210 PRINT " OPTIONS
-----"
220 PRINT ::"1 - PROBLEM 1 "
230 PRINT ::"2 - PROBLEM 2"
240 PRINT ::
250 INPUT LEVEL
260 IF (LEVEL<1)+(LEVEL>2) THEN 2
50
270 DIM A(4,4)
280 CALL CLEAR
290 CALL SCREEN(14)
300 ON LEVEL GOSUB 320,440
310 GOTO 500
320 FOR C=1 TO 4
330 FOR L=1 TO 4
340 A(C,L)=97
350 NEXT L
360 NEXT C
370 FOR T=1 TO 8
380 C=INT(RND*3)+1
390 L=INT(RND*3)+1
400 IF A(C,L)=104 THEN 380
410 A(C,L)=104
420 NEXT T
430 RETURN
440 FOR L=1 TO 4
450 FOR C=1 TO 4
460 A(C,L)=INT(RND*4)+1
470 NEXT C
480 NEXT L
490 RETURN
500 SC=0
510 W$="Q-QUIT"
520 CD=17
530 LI=2
540 GOSUB 1270
550 W$="TRIES="&STR$(SC)
560 CD=20
570 LI=1
580 GOSUB 1270
590 W$="ROW="
600 CD=22
610 LI=1
620 GOSUB 1270

```

```

630 W$="COLUMN="
640 CD=24
650 LI=1
660 GOSUB 1270
670 ON LEVEL GOSUB 1310,1380
680 CALL HCHAR(8,15,134,4)
690 CALL HCHAR(13,15,134,4)
700 CALL VCHAR(8,14,134,6)
710 CALL VCHAR(8,19,134,6)
720 Q=15
730 FOR T=49 TO 52
740 CALL HCHAR(7,Q,T)
750 Q=Q+1
760 NEXT T
770 Q=9
780 FOR T=49 TO 52
790 CALL HCHAR(Q,13,T)
800 Q=Q+1
810 NEXT T
820 CALL HCHAR(22,6,63)
830 CALL SOUND(220,1400,0)
840 GOTO 870
850 CALL SOUND(220,120,0)
860 REM row?,column?
870 CALL KEY(0,K,S)
880 IF S=0 THEN 870
890 IF K=81 THEN 2510
900 IF (K<49)+(K>52) THEN 850
910 LINH=K-48
920 CALL HCHAR(22,6,K)
930 CALL HCHAR(24,9,63)
940 CALL SOUND(220,1400,0)
950 GOTO 970
960 CALL SOUND(220,120,0)
970 CALL KEY(0,K,S)
980 IF S=0 THEN 970
990 IF K=81 THEN 2510
1000 IF (K<49)+(K>52) THEN 960
1010 COLU=K-48
1020 CALL HCHAR(24,9,K)
1030 SC=SC+1
1040 W$=STR$(SC)
1050 CD=20
1060 LI=7
1070 GOSUB 1270
1080 FOR T=1 TO 5
1090 CALL SOUND(1,1000,0,1000*2,
0)
1100 NEXT T
1110 ON LEVEL GOSUB 1440,1850
1120 CALL HCHAR(22,6,32)
1130 CALL HCHAR(24,9,32)
1140 GOTO 820
1150 CALL COLOR(1,6,1)
1160 FOR T=2 TO 4
1170 CALL COLOR(T,13,16)
1180 NEXT T
1190 FOR T=5 TO 8
1200 CALL COLOR(T,9,16)
1210 NEXT T
1220 CALL COLOR(9,16,7)
1230 CALL COLOR(10,16,5)
1240 CALL COLOR(14,2,16)
1250 CALL COLOR(13,8,14)
1260 RETURN

```

```

1270 FOR T=1 TO LEN(W$)
1280 CALL HCHAR(CO,LI+T,ASC(SEG$(W$,T,1)))
1290 NEXT T
1300 RETURN
1310 FOR C=1 TO 4
1320 FOR L=1 TO 4
1330 CALL HCHAR(C+8,L+14,A(C,L))
1340 NEXT L
1350 NEXT C
1360 RETURN
1370 REM print board (2)
1380 FOR C=1 TO 4
1390 FOR L=1 TO 4
1400 CALL HCHAR(C+8,L+14,A(C,L)+48)
1410 NEXT L
1420 NEXT C
1430 RETURN
1440 IF A(LINH,COLU)=97 THEN 1470
1450 A(LINH,COLU)=97
1460 GOTO 1480
1470 A(LINH,COLU)=104
1480 IF LINH=4 THEN 1550
1490 FOR T=LINH+1 TO 4
1500 IF A(T,COLU)=97 THEN 1530
1510 A(T,COLU)=97
1520 GOTO 1540
1530 A(T,COLU)=104
1540 NEXT T
1550 IF LINH=1 THEN 1620
1560 FOR T=LINH-1 TO 1 STEP -1
1570 IF A(T,COLU)=97 THEN 1600
1580 A(T,COLU)=97
1590 GOTO 1610
1600 A(T,COLU)=104
1610 NEXT T
1620 IF COLU=4 THEN 1690
1630 FOR T=COLU+1 TO 4
1640 IF A(LINH,T)=97 THEN 1670
1650 A(LINH,T)=97
1660 GOTO 1680
1670 A(LINH,T)=104
1680 NEXT T
1690 IF COLU=1 THEN 1760
1700 FOR T=COLU-1 TO 1 STEP -1
1710 IF A(LINH,T)=97 THEN 1740
1720 A(LINH,T)=97
1730 GOTO 1750
1740 A(LINH,T)=104
1750 NEXT T
1760 GOSUB 1310
1770 FOR C=1 TO 4
1780 IF A(C,1) <> 97 THEN 1840
1790 IF A(C,4) <> 97 THEN 1840
1800 IF A(C,2) <> 104 THEN 1840
1810 IF A(C,3) <> 104 THEN 1840
1820 NEXT C
1830 GOTO 2270
1840 RETURN
1850 IF A(LINH,COLU)=5 THEN 1860
ELSE 1880
1860 A(LINH,COLU)=0
1870 GOTO 1890

```

```

1880 A(LINH,COLU)=A(LINH,COLU)+1
1890 IF LINH=1 THEN 1960
1900 FOR T=LINH-1 TO 1 STEP -1
1910 IF A(T,COLU)=5 THEN 1920 ELSE 1940
1920 A(T,COLU)=0
1930 GOTO 1950
1940 A(T,COLU)=A(T,COLU)+1
1950 NEXT T
1960 IF LINH=4 THEN 2030
1970 FOR T=LINH+1 TO 4
1980 IF A(T,COLU)=5 THEN 1990 ELSE 2010
1990 A(T,COLU)=0
2000 GOTO 2020
2010 A(T,COLU)=A(T,COLU)+1
2020 NEXT T
2030 IF COLU=1 THEN 2100
2040 FOR T=COLU-1 TO 1 STEP -1
2050 IF A(LINH,T)=5 THEN 2060 ELSE 2080
2060 A(LINH,T)=0
2070 GOTO 2090
2080 A(LINH,T)=A(LINH,T)+1
2090 NEXT T
2100 IF COLU=4 THEN 2170
2110 FOR T=COLU+1 TO 4
2120 IF A(LINH,T)=5 THEN 2130 ELSE 2150
2130 A(LINH,T)=0
2140 GOTO 2160
2150 A(LINH,T)=A(LINH,T)+1
2160 NEXT T
2170 GOSUB 1380
2180 FOR T=1 TO 4
2190 IF A(1,T) <> 0 THEN 2250
2200 IF A(2,T) <> 0 THEN 2250
2210 IF A(3,T) <> 0 THEN 2250
2220 IF A(4,T) <> 0 THEN 2250
2230 NEXT T
2240 GOTO 2270
2250 RETURN
2260 CALL SCREEN(14)
2270 CALL CLEAR
2280 CALL SCREEN(11)
2290 T=2000
2300 FOR P=1 TO 2
2310 K=-143*(P=1)-32*(P=2)
2320 FOR N=1 TO 12
2330 CALL SOUND(-200,T-600,0,T-570,2)
2340 T=T-30
2350 CALL HCHAR(13-N,17-N,K,N*2)
2360 CALL VCHAR(13-N,16+N,K,N*2)
2370 CALL HCHAR(12+N,17-N,K,N*2)
2380 CALL VCHAR(13-N,17-N,K,N*2)
2390 NEXT N
2400 NEXT P
2410 PRINT "YOU SOLVED THE PROBLEM IN"
2420 PRINT :TAB(7);SC;"TRIES"
2430 PRINT ::TAB(3);"TYPE RUN FOR ANOTHER PLAY"
2440 FOR T=1 TO 2000
2450 NEXT T
2460 END

```

◻ HIGH SCORE ◻

ALPINER	110,392*
ATTACK	388,800
BIGFOOT	34,940
BLASTO	47,340
BUCK ROGERS	295,349*
BURGERTIME	500,000
CAR WARS	56,250
CHISHOLM TRAIL	89,990
CONGO BONGO	250,000
DEFENDER	3,720,000
DONKEY KONG	1,741,300
FATHOM	5,881
FROGGER	12,620*
HOPPER	60,140
INVADERS	54,212
JAWBREAKER	15,870
JUNGLE HUNT	32,160
M*A*S*H	4,027
MICROSURGEON	4,253
MINER 2049er	54,655
MOON MINE	131,720
MOON PATROL	133,150
MS PACMAN	151,610
MUNCHMAN	458,580
MUNCHMOBILE	27,290
PACMAN	43,180
PARSEC	10,842,800
PICNIC PARANOIA	612,520
POPEYE	110,700
POPEYE	76,400
PROTECTOR II	125,764
Q*BERT	24,700*
SHAMUS	104,790
SLYMOIDS	113,905
SNEGGIT	2,635
STAR TREK	1,097,000
SUPER DEMON ATTACK	8,750
TOMBSTONE CITY	2,135,950
TREASURE ISLAND	26,100

★ NEW HIGH SCORERS ★

DAVID CAINE ON Q*BERT
 DAVID CAINE ON FROGGER
 JOSE DORAN ON ALPINER
 JOSE DORAN ON BUCK ROGERS

◻ TOP FIVE ◻

5	5	5	5	5	5	5
GAMES 5						
POLE POSITION						
ADVENTURE PIRATE						
MOON PATROL						
ALPINER						
JUNGLE HUNT						
5	5	5	5	5	5	5
5	5	5	5	5	5	5
TAPES 5						
FUN PAC 2						
GAMES WRITER 2						
GAMES WRITER 1						
EXCALIBUR						
LIST FILE						
5	5	5	5	5	5	5
5	5	5	5	5	5	5
OFFICE 5						
MINIWRITER III						
P.R.K.						
STATISTICS						
MINI MEMORY						
HOME BUD MANAGEMENT						
5	5	5	5	5	5	5
5	5	5	5	5	5	5

SEND YOUR HIGH SCORES TO
 The Editor
 PARCO MAGAZINE